DM 7,- Sfr 7.- Ös 56.- März ´89 Nr. 3

AMIGA

KICK START

COMPUTER MAGAZ

JANUS

Die zwei Gesichter des AMIGA

R.C.T.

Requester Construction Tool

AMIGA-NEWS

Das plant Commodore

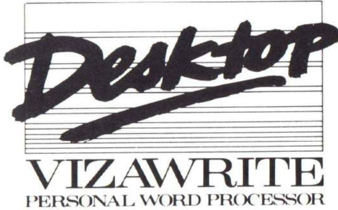
FLOPPY-KURS

Geheimnisse der schwarzen Scheibe

Superback • Zeichenkiste • Proportionaldruck
Spiele • Modem Talking • KICKUP • NEWS

AMIGA-NEWS 3/89

Ein Informationsservice von Ihren Distributoren DTM Deutschland und MICROTRON Schweiz



VIZAWRITE POWER
High-Quality Amiga-Fonts

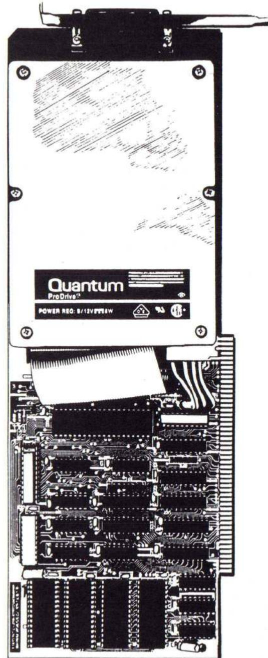
Glücklich ist wer einen Drucker mit verschiedenen Schriftsätzen sein Eigen nennt. Aber auch der Glückliche sucht immer wieder nach Abwechslung.
Vizawrite Desktop bietet Ihnen die Möglichkeit, die breite Palette Amiga-Fonts in Auflösungen von 120, 180, 240 und 360 dpi auszudrucken. Wie geht das?
Die Bildschirm Fonts haben im Fonts-Verzeichnis einen "großen Bruder", der die Zeichen breiter und höher gezeichnet enthält. Beim Ausdruck werden diese Zeichensätze vom Programm gelesen und dann entsprechend der zu druckenden Größe komprimiert. Dadurch wird logischerweise eine höhere Dichte erreicht, da für den Druck wesentlich mehr Einzelpunkte zur Verfügung stehen. Besondere Voraussetzung sollte ein 24-Nadel- oder Laserdrucker sein.
Auf der Systemdiskette befinden sich beim Kauf bereits 4 verschiedene dieser HQ (High-Quality) Fonts.
Wer zusätzliche Amiga Fonts mit hÖherer Auflösung einsetzen möchte, kann ohne weiteres mit einem Font-Editor, z.B. Calligrapher (mit deutschem Handbuch bei uns erhältlich), eigene Zeichensätze erstellen.
Die mit Vizawrite Desktop zusätzlich mitgelieferte Fontdiskette enthält z.Zt. die vier HQ-Fonts und 20 weitere Amiga Fonts in der normalen Auflösung.
Ihrer gestalterischen Kreativität beim Schreiben von Dokumenten sind somit kaum Grenzen gesetzt, allerdings sollten Sie den Grundsatz der druckenden Zunft im Auge behalten und nicht mehr als 2-3 Fonts pro Seite verwenden.
In der nächsten Ausgabe berichte ich Ihnen über die Serienbrieffunktion in Vizawrite.
Ihr V. Write

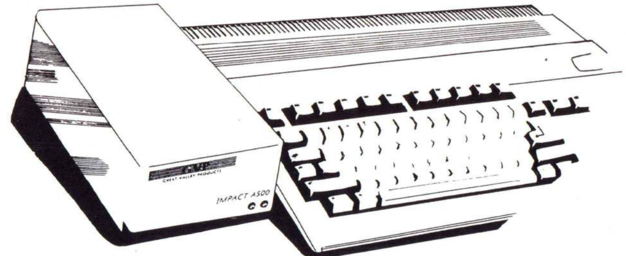
DOS TO DOS

ist ein nützliches Programm zum Datenaustausch zwischen verschiedenen Diskettenformaten z.B. Amiga und PC oder Atari. DOS TO DOS kann auf Amiga Laufwerken Disketten im MS-DOS Format lesen, schreiben und formatieren. So wird der Datentransfer zum Kinderspiel.
DOS TO DOS wird mit einer ausführlichen deutschen Anleitung geliefert und kostet 128,- DM.

SCSI High-Speed Autoboot Festplatten

- Hohe Datenübertragungsrate durch DMA Transfer zwischen Drive & Controller
- Keine Verzögerung von DMA Prozessen, z.B. Animation oder Musiksampling
- Autoboot ab Kickstart 1.3 direkt von Fast-File Partition möglich
- Autoboot einer vorhandenen XT- oder AT-Erweiterung über PC Partition
- Deutsche Installationssoftware mit Workbench 1.3 und Fast-File System
- Zusätzlicher RAM Ausbau bei A-2000 SCSI Controller und Amiga 500 Drive
- Durchgeschleifter SCSI-Busstecker für 7 weitere SCSI-Einheiten





NEU! GVP Amiga 500 Festplatten

GVP SCSI Drive 20 MByte 28 ms.	1798,- DM
GVP SCSI Hardbox 46 MByte 28 ms.	2495,- DM
GVP Autoboot/Ram Karte 2 MB opt.	179,- DM

GVP Amiga 2000 Festplatten

GVP SCSI Hardcard 20 MB 28 ms.	1498,- DM
GVP SCSI Hardcard 30 MB 28 ms.	1698,- DM
GVP SCSI Hardcard 46 MB 28 ms.	1998,- DM
GVP SCSI Hardcard 40 MB 11 ms. **Quantum**	2298,- DM
GVP SCSI Hardcard 80 MB 11 ms. **Quantum**	3298,- DM
GVP SCSI Controller, 2 MB Fast Ram opt.	698,- DM

Quantum Festplatten gelten zweifelsohne in Kennerkreisen als Porsche unter den Platten. Eine außerordentlich schnelle Zugriffszeit von 19 msec., die sich durch den integrierten 64 KByte Read-Ahead Cache bei geschickter Ausnutzung bis auf 11 msec. steigern läßt, das patentierte Autoparksystem (Airlock) und die fast doppelt so hohe Lebensdauer (MTBF 50.000 Stunden) gegenüber herkömmlichen Festplatten sichern Ihnen jahrelanges, zuverlässiges Arbeiten.

Unsere Bestseller

Z.Zt. führen wir Produkte der Firmen AEGIS, AMERICAN COVERS, ANAKIN RESEARCH, BYTE BY BYTE, CENTRAL COAST SOFTWARE, DTM, ELECTRONIC ARTS, IDEAS CREATED, LATTICE, METACOMCO, MICRO WAY, MINDWARE INTERNATIONAL, PRECISION SOFTWARE, GREAT VALLEY PRODUCTS, VIZA SOFTWARE. Hier eine Auswahl der derzeitigen Bestseller:

AEGIS alle Produkte	lieferbar
Animate 3D	298,- DM
Calligrapher	229,- DM
Digi View Gold deutsch	398,- DM
Digi Paint deutsch	148,- DM
DISCovery deutsch	198,- DM
Disk to Disk	109,- DM
DOS to DOS deutsch	128,- DM
EASYL Zeichentabletts	
Easyl Amiga 500	698,- DM
Easyl Amiga 1000	798,- DM
Easyl Amiga 2000	898,- DM
Funktion deutsch	98,- DM
Jitter Rid Filterscheibe	29,95 DM
Pageflipper deutsch	77,- DM
Pageflipper F/X plus	398,- DM
Quarterback deutsch	128,- DM
Sculpt/Animate 4D	998,- DM

HANDBÜCHER einzeln:

AEGIS Audiomaster	29,95 DM
AEGIS Sonix	39,95 DM
AEGIS Videotitler (4/89)	
Balance of Power	29,95 DM
Calligrapher	29,95 DM
Comicsetter	29,95 DM
Flugsimulator II	29,95 DM
Jet	29,95 DM
Kampfgruppe	29,95 DM

INTERLACE OHNE FLIMMERN!

ist kein Problem mit dem Flicker Fixer von Micro-Way. Diese spezielle Grafikkarte puffert die vom Amiga erzeugten Bilder und gibt diese flimmerfrei auf einem Multisync Monitor wieder. Die Installation ist denkbar einfach, Handbuch und Diskette gehören zum Lieferumfang. Flicker Fixer kostet 1298,- DM.

AmiEXPO New York

auch dieses Mal sind wir wieder vor Ort dabei. Auf der wohl derzeit größten Ausstellung mit 'pur Amiga' werden wieder alle Amiga-Anbieter mit Rang und Namen vertreten sein. Seien Sie gespannt auf die neuen Produkte, über die wir in der nächsten Ausgabe berichten werden.
Übrigens wird von der Messeleitung z.Zt. eine Ami-Expo in Deutschland geplant. Der Termin liegt noch nicht fest, es wird von Herbst gesprochen. Näheres in der nächsten Ausgabe der News.

Preissenkungen

bitte beachten Sie die ab sofort gültigen Verkaufspreise:

Lattice 5.0	598,- DM
Jitter Rid	29,95 DM
Aegis DIGA!	128,- DM

DTM Restposten

Arazoks Tomb	49,- DM
Butcher 2.0 PAL dtsch.	98,- DM
Fast Lightning	39,- DM
Grabbit	49,- DM
Garrison I / II	39,- DM
Spaceballer	10,- DM
Metacomco Toolkit	98,- DM
Bad Cat	39,- DM

KOSTENLOSE INFORMATIONEN

senden wir Ihnen gerne zu allen Neuheiten und wenn Sie meinen, dies sei alles, bekommen Sie noch unsere Gesamtliste mit über 600 Artikeln. Einfach diesen Coupon auf eine Postkarte kleben und an DTM bzw. in der Schweiz an MICROTRON senden.

O Senden Sie mir Ihre Gesamtliste AMIGA Soft- und Hardware.
O Senden Sie mir detaillierte Informationen zu folgenden Produkten:

Alle unsere Produkte erhalten Sie in gutsortierten Fachgeschäften, die wir Ihnen gerne nennen sowie unseren DTM-Computershops in Wiesbaden. Selbstverständlich können Sie alle Produkte auch direkt schriftlich oder telefonisch bei DTM bzw. MICROTRON bestellen.

Alle Preise sind unverbindlich empfohlene Verkaufspreise. Preisänderungen und Irrtümer vorbehalten. Es gelten unsere AGB.

MICROTRON
COMPUTERPRODUKTE
Postfach 69 Bahnhofstr. 2
Tel. 032 872429 Fax 032 872482
CH-2542 PIETERLEN

Poststraße 25
6200 Wiesbaden-Bierstadt
(0 61 21) 56 00 84
fax (0 61 21) 56 36 43

Werbung und EDV GmbH

Es war einmal vor langer, langer Zeit ein Ort in einem Tal, das von der Wärme der Sonne verschont wurde, in dem die Flüsse reines und kristallklares Wasser trugen, wo die Blumen die skurile Farbe der Atmosphäre angenommen hatten und das von Bergen eingeschlossen war, die so riesig waren, daß man ihr Ende nur ahnen konnte. Dieser wurde von einem Volk bewohnt, das es heute nicht mehr gibt. Die Frauen waren groß und mächtig, die Männer im Gegenteil schmale Gestalten, die kaum Schatten warfen. Ihre Essensgewohnheiten waren sehr seltsam und ihre Götter sehr vielfältig.

Drei junge Zauberer hatte dieses Volk, und diese waren ehrgeizig und sehr erfinderisch. Sie hatten der Kunst der Verwandlung und anderen Alchemien ihr ganzes Leben geopfert. Die ganze Zeit meditierten sie oder verwandelten alles, was die Augen sehen oder der Instinkt riechen konnte. Die Tage der Meditation waren für das Volk ohne jegliche Anregung, aber die Tage, an denen die Verwandlungskunst praktiziert wurde...! Diese Tage waren besonders schön, weil man in jeder Ecke, in jeder Falte ihre Kunst bewundern konnte. Eine fast nicht wahrnehmbare Bewegung einer Hand und der schwere, liegende Stein verwandelte sich in einen wunderschönen Kolibri, der schnell davonflog. Ein Zucken mit der oberen Lippe und es begann zu regnen, sieben lange Tage und sieben lange Nächte. Besonders gut waren sie in der Erzeugung von Sonne und Mondfinsternis sowie bei Levitationen aller möglichen Gegenstände. Die drei Zauberer waren sehr zufrieden mit ihrem Können und fühlten sich sehr wohl unter ihrem Volk, bei dem sie großes Ansehen genossen und nichts zu tun brauchten. Aber das Leben der Wohlhabenden, Gelehrten und Weisen ist ohne jegliche Herausforderung langweilig, und deswegen entschlossen sie sich eines Tages, die Kunst der Umwandlung an sich selbst zu praktizieren.

Der älteste unter ihnen, der sich die Elemente des Wassers und die Farbe Schwarz unterworfen hatte, dachte, daß die größte Verwandlung, die er an sich selbst vornehmen könnte, die maximale Vergrößerung seines Kopfes sei, um dadurch sein Denkvermögen so zu erweitern, daß er das ganze Wissen der Menschheit aufnehmen kann. Dreimal krähte der Hahn, bevor er fertig war. Die beiden anderen vergaßen ihre Meditationsübungen und Zeit für Verwandlungsexperimente gab es auch nicht mehr. Das ganze Volk machte nichts anderes als zu warten. Als der älteste von den drei Schwankenden wie ein Säufer aufgestanden war, und seine Hütte verlassen wollte, merkte er, daß es ihm nicht möglich war, sein Gleichgewicht zu halten. Zu schwer war sein Kopf geworden, und so groß, daß er seine Hütte nicht mehr verlassen konnte. Bei seinem Versuch, hinauszugehen, stieß er mit dem Kopf gegen die Wand und bekam Kopfschmerzen. Und weil sein Kopf so groß war, waren seine Schmerzen noch größer. Als die anderen zwei Zauberer das tragische Schicksal ihres Kameraden sahen, zogen sie sich zurück und meditierten.

Mehrmals gingen die mächtigen Frauen und die schmalen Männer auf die Felder. Mehrmals tranken die Kinder aus den tiefen Fässern den Löwenzahn-Wein, bevor sich der Zweitälteste entschloß, eine Verwandlung an sich zu bewirken. Feuer und Sulfur waren seine Elemente. Die Göttin des Mondstaubs war seine Divinität. Er dachte, daß die größte Gabe, die die Menschheit besaß, die Hände seien. Es war die Entwicklung dieser Extremitäten, die den Menschen langsam, aber sicher von dem Affen trennte. Die Entwicklung der Intelligenz kam später. Dachte der Zweitälteste. Man sollte die Füße in Hände verwandeln, denn dadurch hätte man größere Möglichkeiten etwas anzufassen, und mit vier Hände gleichzeitig zaubern zu können. Viermal wurde die große Wassergrube geleert und viermal neu gefüllt, bis die Wandlung vollbracht war. Der jüngere Zauberer wartete unruhig am Eingang der Hütte, und unter dem ganzen Volk herrschte ein unerträgliches Schweigen. Der zweitälteste Zauberer sah seine verwandelten unteren Extremitäten und fand sie schön. Dann versuchte er, Schuhe anzuziehen, aber es ging nicht. Nach einer Weile gab er es auf, und dachte, es werde auch ohne gehen. Er stand auf, und ging mit einem unguten Gefühl hinaus. An der Schwelle der Hütte stolperte er und rollte vor den Augen des jüngsten Zauberers und dem versammelten Volk die Treppe hinunter. Es gelang ihm aber trozdem noch, aufzustehen. Die Steine, die den Weg füllten, schmerzten seine unteren Hände so sehr, daß er immer wieder aufspringen mußte. Jede dieser unkontrollierten Bewegungen verursachte eine Verwandlung im Volk. So wurde der Bürgermeister ungewollt in ein rosa Schwein verwandelt und die stolze Fahne, die immer gegen den Wind ankämpfte, in einen weißen Adler. Panik brach unter dem Volk aus, das nichts anderes im Kopf hatte, als schnell davonzulaufen, um Körper und Seele zu retten. Nach einer Weile setzte sich der Zweitälteste hin und weinte. Der jüngere Zauberer trug ihn zu seiner Hütte und schloß nachdenklich die Tür hinter sich.

Erneut gingen die Männer und die Frauen auf die Felder. Die Hausdrachen bekamen Nachwuchs und die Schlangen wechselten mehrmals ihre Haut. Der jüngere Zauberer war vorsichtig geworden, aber trotzdem entschloß er sich ebenfalls zu einer Verwandlung. Die Luft war sein geliebtes Element und Kassandra, die Göttin, die auf den Sonnenstrahlen galoppiert, seine Divinität. Er dachte, man müßte zwei Gesichter haben. So könnte man mehr sehen, und zwei Gesichtsausdrücke gleichzeitig haben, wie Janus, der altrömische Gott, und niemand wüßte genau, was man denkt. Fünfmal heulte der alte Wolf den Vollmond an, bis die Umwandlung fertig war. Kein Zauberer stand mehr vor der Hütte und das Volk hatte sich schon lange irgendwo anders niedergelassen. In zwei Spiegeln betrachtete der jüngste Zauberer seine zwei Gesichter. Das eine war sein bisheriges Gesicht, und das andere ebenfalls, nur in anderer Richtung. Mit dem einen lächelte er, mit dem anderen weinte er. Mit dem einen machte er die Augen ganz auf und mit dem anderen ganz zu. So spielte er eine ganze Weile, bis er sich entschloß, die Hütte zu verlassen. Draußen lag Schnee und die kalte, grelle Sonne stach in seine vier Augen. Er versuchte, sein Umfeld und die Gegenstände, die um ihn herum lagen, einzeln zu betrachten. Es gelang ihm aber nicht, sich zu konzentrieren. Eine Lawine von Visionen und Lichteinflüssen füllte sein Gehirn immer mehr. Etwas später begann er sich zu drehen und bewegte sich wahllos in verschiedene Richtungen. Erst rannte er in die eine und dann in die andere. Plötzlich ging er in den Wald hinein und kam nie wieder.

Das letzte Mal als ich ihn gesehen habe, saß er völlig nackt, ein Gesicht zur Wand und das andere auf mich gerichtet. Wie gesagt: Es war einmal vor langer, langer Zeit...

M. Merino

INHALT

AKTUELLES

SOFTWARE

HARDWARE

GRUNDLAGEN

JANUS

Die zwei Gesichter des AMIGA

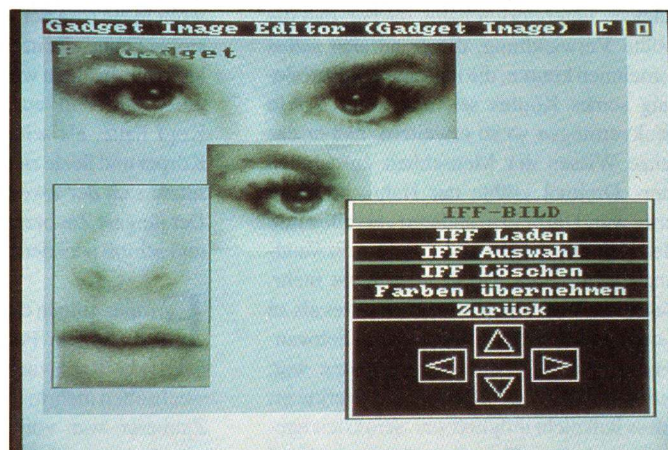
Die neue Software für die AT-Karte kann auch für das PC-Bridgeboard und die Sidecar verwendet werden. Wie die Software zu bedienen ist, was man beachten muß, welche Bugs noch existieren, und was sie leistet, lesen Sie ab

Seite 26

Intuition

leicht gemacht. Mit dem Requester Construction Tool können Sie schnell und einfach eine professionelle Benutzeroberfläche erstellen. Wie einfach das ist, erfahren Sie ab

Seite 100

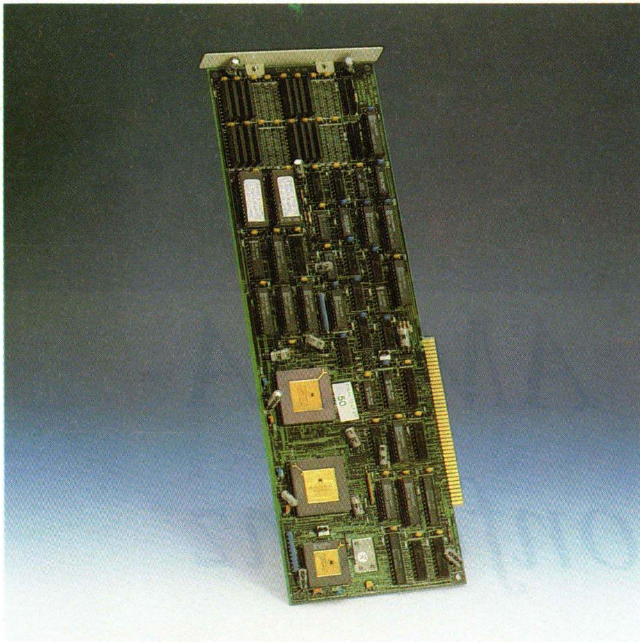


3 '89

Commodores 68020-Karte

Die 68020-Karte von Commodore weist einige Besonderheiten gegenüber anderen 68020-Karten auf. Nicht nur, daß sie als einzige Karte dank eines speziellen Chips kompatibel zum UNIX-Betriebssystem ist, nein, auch andere Besonderheiten sind ihr gegeben.

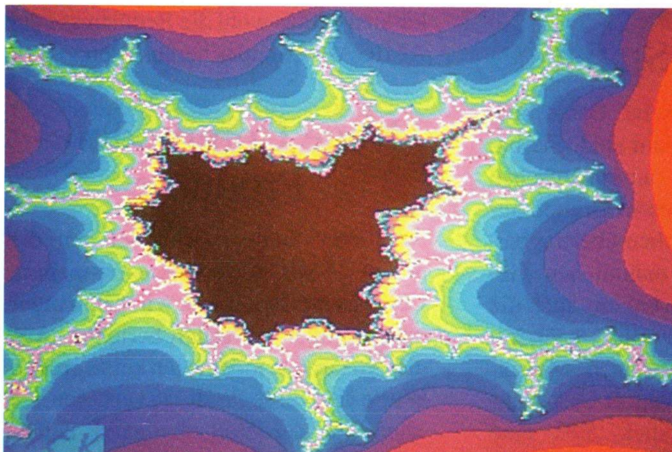
Seite 17



Fraktalgeneratoren

besitzen einen enormen Reiz, sind solche Generatoren doch eine typische Computeranwendung. Der FRAKTALGENERATOR 3D erlaubt Berechnungen in 3D und hebt sich nicht nur dadurch von der Masse der rechenintensiven Programme ab, er bietet auch einige weiterführende Features. Welche das sind? Ab

Seite 54



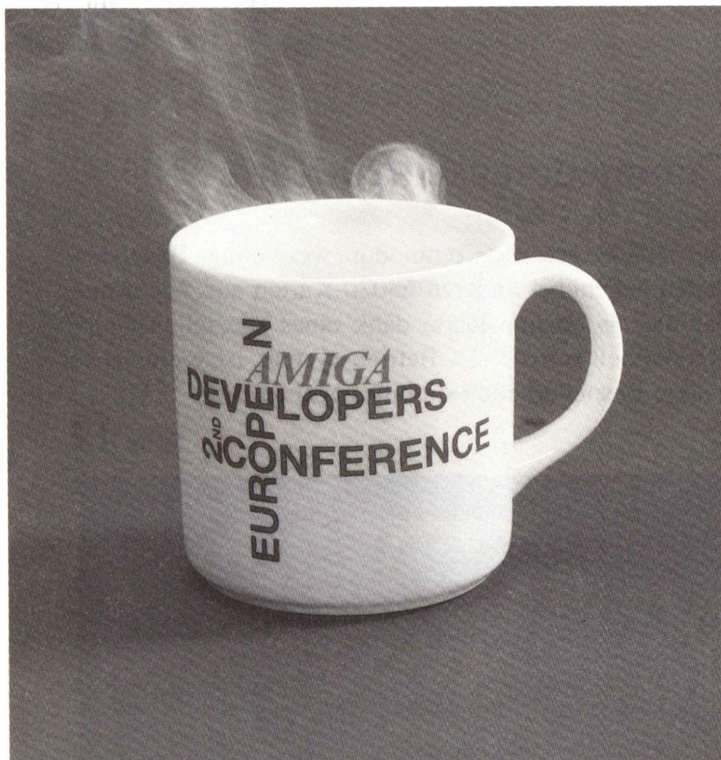
LISTING

KICKS FÜR INSIDER

SPIELE

RUBRIKEN

Brandheiße Neuigkeiten rund um den AMIGA



Zweite europäische AMIGA-Entwicklerkonferenz

Die zweite europäische Entwicklerkonferenz fand diesmal im Frankfurter Intercontinental Hotel statt. Knapp 300 Entwickler hatten sich angesagt, um den neuesten Stand der AMIGA-Entwicklungen in Erfahrung zu bringen. Ein weiterer Grund, der Konferenz beizuwohnen, war mit Sicherheit, daß fast alle wichtigen AMIGA-Entwickler aus Amerika und Europa im Bereich Soft- und Hardware ihr Kommen angekündigt hatten. So konnten sie wichtige Fragen direkt an den (die) betreffenden Entwickler stellen.

In fast 30 Vorträgen wurden die verschiedensten Themen behandelt, angefangen mit dem neuesten IFF-Standard über die neue KICKSTART-Version 1.3 (1.4) bis hin zu Transputern. Der einfache AMIGA-Anwender wird mit Sicherheit mit den meisten Vorträgen wenig anfangen können, für Entwickler konnten aber einige Probleme beiseitegeräumt werden. Für Sie haben wir das Interessanteste und das, was uns in naher Zukunft erwartet, zusammengefaßt.

Ein wenig Statistik

Zuvor einige statistische Daten, die mitunter recht interessant sind. Sie beziehen sich auf den amerikanischen Markt und sind aus diesem Grunde nur bedingt auf den deutschen übertragbar. 94% der User sind männlichen Geschlechts (daran dürfte bei uns auch die MISS AMIGA-Wahl nicht geändert haben), 65% besitzen einen Matrix-Drucker (35% einen Matrix-Farbdrukker), 65% einen Joystick, 59% eine Speichererweiterung, 9% eine Festplatte, 8% ein Midi-Interface, 69% ein Zweitlaufwerk. 43% spielen mit dem AMIGA, 31% nutzen eine Textverarbeitung, 23% programmieren, 22% nutzen Animationsprogramme, 21% Public Domain-Software, 19% Musikprogramme und 18% Datenbanken. Eine ganze Reihe von weiteren statistischen Daten ließe sich noch anführen; doch ich möchte es darauf beruhen lassen, da die Zahlen nur bedingt auf den deutschen Raum zu übertragen sind. Entwickler können daraus wohl die Bedürfnisse der AMIGA-Anwender herauslesen.

Neue Commodore-Produkte

An dieser Stelle möchte ich auf die neuen AMIGA-Produkte eingehen. Über die meisten haben wir schon mehr oder weniger ausfürlich berichtet. Aus diesem Grunde möchte ich keine großen Worte über diese Produkte verlieren, sie seien nur der Vollständigkeit halber erwähnt.

A2286 Bridgeboard:
die lang ersehnte und endlich erhältliche AT-Karte

A2300 Genlock:
das Genlock für den Videoslot des AMIGA 2000, bereits ausführlich getestet in Heft 1.89

A2620 68020 Card:
die 69020 Karte mit 68881-FPU und MMU für den Prozessorslot des AMIGA 2000.

A2090A-Harddisk-Controller:
Festplattenkarte für den AMIGA 2000

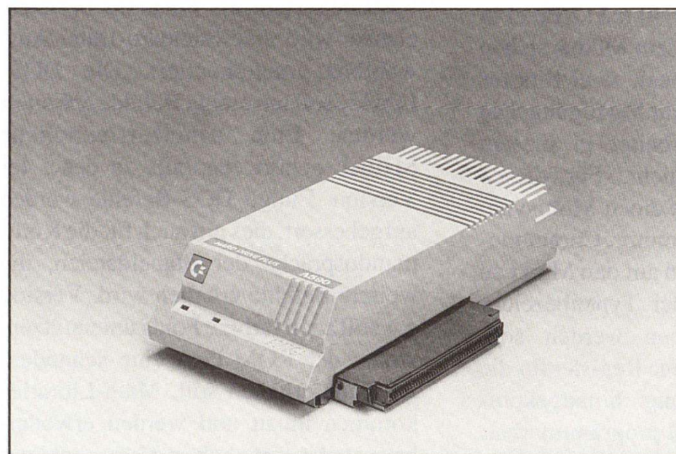
A2058-Memory-Expansion-Card:
Die Speichererweiterung kann bis maximal 8 MByte aufgerüstet werden, wobei die Aufrüstung mit MBits-Chips in MByte-Schritten erfolgen kann.

A2024-HiRes-Monochrome-Monitor:
Drei Non-Interlaced-Modi unterstützt der Monitor - 1008 x 1024, 704 x 512, 704 x 256. Der Monitor arbeitet ohne Änderungen an der Hardware mit allen AMIGA-Modellen. Es sind jedoch 1 MByte RAM benötigt. Zum monochromen Monitor möchte ich noch ein paar Worte verlauten lassen. Vier Graustufen werden unterstützt - Schwarz, Hell-, Dunkelgrau und Weiß. Für die höchste Auflösung, 1008 x 1024, wird eine spezielle Workbench ausgeliefert die den Modus unterstützt. Verständlich ist, daß die höchste Auflösung nur von sehr wenigen Programmen unterstützt wird, alle anderen werden z.Z. schon von vielen Programmen genutzt.

A2350-Professional-Video-Adapter:
Etwas ausführlicher möchte ich den unter der Typenbezeichnung A2350 laufenden Professional-Video-Adapter für den A2000 beschreiben. Der Adapter beinhaltet zwei Steckkarten und ein Software-Paket. Auf den Karten befinden sich ein Genlock und ein Freeze Frame, der es in Echtzeit erlaubt, Videobilder in Zusammenarbeit mit einem Digitizer in den Videospeicher des AMIGA zu digitalisieren. Der Digitalisierer unterstützt alle AMIGA-Auflösungen bis hin zum HAM-Modus mit 4096 Farben. Die Software erlaubt ein Umschalten der Videoquellen, Schärfe und Farbe können ebenfalls eingestellt werden.

Für den Festplattenkontroller A2090 wird des weitereren eine kleine Erweiterung erhältlich sein, die es ermöglicht, direkt von Festplatte zu booten. Die A2090B beinhaltet zwei zusätzliche EPROMS, die das Booten unter Kickstart 1.3 ermöglichen. Wie auch die A2090 kann die Karte in einen freien Amiga-Slot gesteckt werden. Das große Manko der Autoboot-Festplatte besteht darin, daß zwei wertvolle Steckplätze für eine Festplatte 'flöten'gehen.

Commodore hat den AMIGA 500 nicht vergessen, das A590 Gerät beinhaltet eine 20 Mbyte-Festplatte und eine 2 MByte-Speichererweiterung. Die Platte wird an den Expansionport des A500 gesteckt. Das Interface ist mit einer DMA (Direct Message Adapter) versehen. Die durchschnittliche Zugriffszeit liegt bei 80ms. Der Speicher kann nach und nach aufgerüstet werden.



A590, für den AMIGA 500 ist die Festplatte mit integriertem Speicher gedacht.

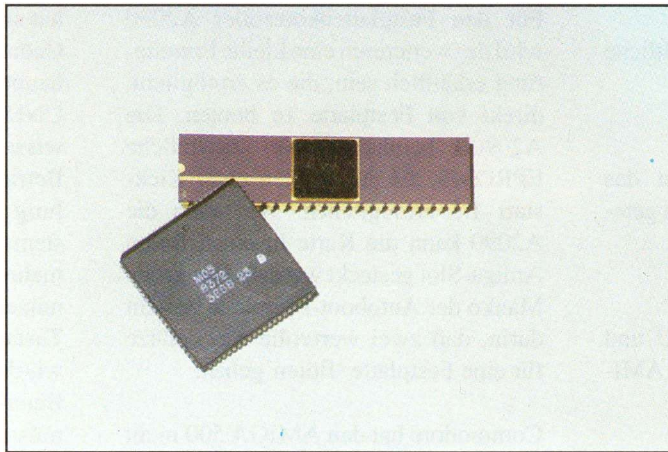
Neue serielle Karte

Völlig neu ist eine serielle Karte für den A2000. Die A2232 beinhaltet 7 serielle Schnittstellen, wobei jede einzelne getrennt mit einer Übertragungsrate von 19.2 KBaud betrieben werden kann. Auf der Karte befindet sich ein Verwandter des C64-Prozessors, der 6502. Er kann vom 68000-Prozessor des AMIGA angesteuert und programmiert werden. Ein AMIGA-Slot geht freilich 'flöten' für die sieben Schnittstellen. Viele AMIGA-Anwender werden sich zu recht nach dem Nutzen der Karte fragen, benötigen sie doch lediglich den bereits vorhandenen für ein Modem, in den seltensten Fällen vielleicht noch für einen seriellen Drucker. Commodore hat sich bei der Entwicklung aber seine Gedanken gemacht, und die Karte hauptsächlich für den Einsatz unter UNIX entwickelt. Wie Sie vielleicht wissen, ist UNIX ein Multiuser-Betriebssystem (eine Weiterentwicklung des Multitasking-Betriebssystems). Es können demnach gleichzeitig mehrere Benutzer dieselbe Hardware nutzen. Da es einiges Gedränge an der Tastatur des UNIX A2000ers geben würde, wenn fünf, sechs oder sieben Benutzer ihre Befehle eingeben würden, müssen sogenannte Terminals bereitstehen. Der A500 ist beispielsweise als Terminal einsetzbar. Über die serielle Schnittstelle der A2232-Karte kann dann der A500 mit dem UNIX-Hostrechner verbunden werden. Rechnerisch sind mit der Karte 8 Terminals anschließbar, sieben mit Hilfe der Karte, und eines kann mit dem bereits serienmäßigen Port verbunden werden. Das Thema UNIX greife ich ein wenig später noch einmal auf, Commodore hat einige interessante Ankündigungen vorgenommen.

1 MByte Chip-Mem

Kommen wir zu einem Thema, das die AMIGA-User schon lange beschäftigt, die neuen Customchips. Bei Commodore wird die Chipsammlung kurz ECS (Enhanced Chip Set) genannt. Das ECS enthält die drei neuen Customchips Denise, Fat Agnus und Gary. Was wir schon lange wissen, ist, daß sie in der Lage sein sollen, 1 MByte ChipMem zu adressieren. Dadurch können endlich komplexere Grafiken erstellt und längere Sounds gesamplet werden. Eine Adressierung bis 2 MByte ist für Maschinen in ferner Zukunft geplant. Des weiteren kommt ein neuer Hires-Mode dazu, genannt Superhires. Er bringt eine Auflösung von 640 x 512 Pixel in Non-interlace, also flimmerfrei. Der Agnus- und Denise-Chip sind erheblich erweitert worden, ich möchte Ihnen die neuen Eigenschaften nicht vorenthalten.

Der neue Fat Agnus



Die neuen Customchips Fat Agnus und Denise werden nur unter der Betriebssystem-Version 1.4 einwandfrei arbeiten. Sie ermöglichen einen flimmerfreien Superhires-Modus mit 640 x 512 Pixel und einer Adressierung von 1 MByte Chipmem.

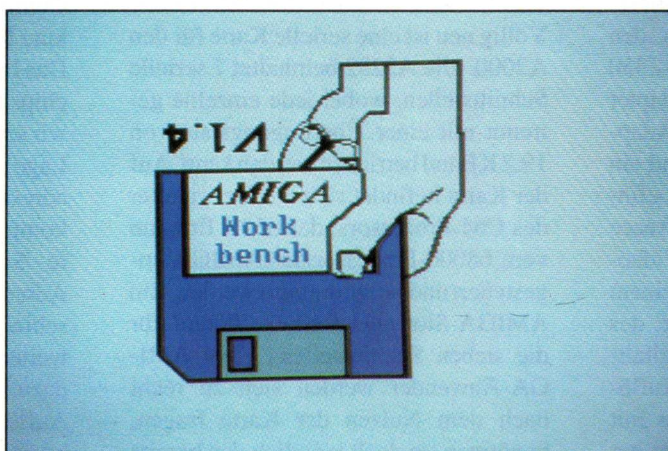
Wie schon erwähnt, ist der erweiterte Agnus in der Lage, 1 MByte ChipMem zu adressieren, des weiteren können mit dem Chip Frequenzen bis zu 28.86 KHz gesamplet werden. Ein neues Blitter-Register, das nur die unteren 8 Bytes schreibt, ist ebenfalls hinzugekommen. Dadurch kann mit geschickter Programmierung die Software enorm beschleunigt werden. Kommen wir zum Customchip Denise.

Der neue Denise

Der Denise ist der Grafikmanager des AMIGA, also auch zuständig für den neuen Grafikmodus. Superhires bringt eine Auflösung von 640 x 512 Pixel in Non-Interlace. In diesem Modus stehen 16 Farben (4 Bitplanes), zwei Bits für Rot, Grün und Blau, zur Verfügung. Die Kollisionsabfrage arbeitet in diesem Modus allerdings nicht. Für diesen Modus benötigen Sie einen Multisync- oder einen Bisync-Monitor. Commodore plant, einen solchen auf den Markt zu bringen, der unter der Typenbezeichnung 1950 vertrieben werden soll. Ebenso sind einige neue Register für die horizontale Auflösung hinzugekommen, damit ist sie voll programmierbar. Ein Extra Half-Bride-Modus ist ebenfalls im neuen Denise integriert. Wohlgemerkt ist das ECS nur unter der Betriebssystem-Version 1.4 lauffähig, die z.Z. mitten in der Entwicklung steht.

Kickstart 1.4

Obwohl gerade erst die neue Betriebssystemversion 1.3 fertiggestellt und bei den Händlern erhältlich ist, arbeiten Commodores Entwickler fleißig an der Version 1.4. Was die neue Kickstart 1.4 und Workbench 1.4 bringen werden, ist schon eine ganze Menge. Die Änderungen zwischen der Version 1.1 und 1.2 bzw. 1.3 sind im Vergleich zu den denen zwischen 1.3 und Version 1.4 gering. Einer der wichtigsten Aspekte ist, daß das ECS unterstützt wird. Des weiteren wird der monochrome Monitor A2024 ansteuerbar sein. Die Intuition-Library wird ebenfalls geändert, genauer gesagt, die Systemgadgets. Die Preferences und die Workbench-Oberfläche werden auch einige Manipulationen erfahren. Das AMIGADOS wird das schnellere FastFileSystem als Standard besitzen, ebenso wird eine Standard-Datei-Auswahlbox implementiert. Die DOS-Library soll um einige Befehle erweitert werden. Eine benutzerfreundliche SHELL gehört ebenfalls zu der 1.4er Version. Einige DOS-Befehle werden aufgebessert, dies gilt auch für die Kommandosprache der Stapeldateien, die weitere Befehle erhalten wird. Version 1.4 soll auch farbige Fonts unterstützen, wobei der ROM-Font eine schnellere Ausgabe erhalten soll. Math-Libraries kommen hinzu und werden erweitert bzw. verbessert. Ach ja, Gurus gehören unter der 1.4-Version der Vergangenheit an, stattdessen werden einfache Fehlermeldungen ausgegeben.



An der neuen Betriebssystem-Version 1.4 wird fleißig entwickelt, bis zur Vollendung wird der Mond aber noch ein paar Mal auf und wieder unter gehen.

Wie Sie eben lesen konnten, ist einiges geplant für die Version 1.4. Bis zur endgültigen Fertigstellung wird aber mit Sicherheit noch einige Zeit vergehen. Ein bis zwei Jahre sollten eingeplant werden, bedenkt man, daß viele Features der Version momentan nur auf dem Reißbrett existieren.

UNIX

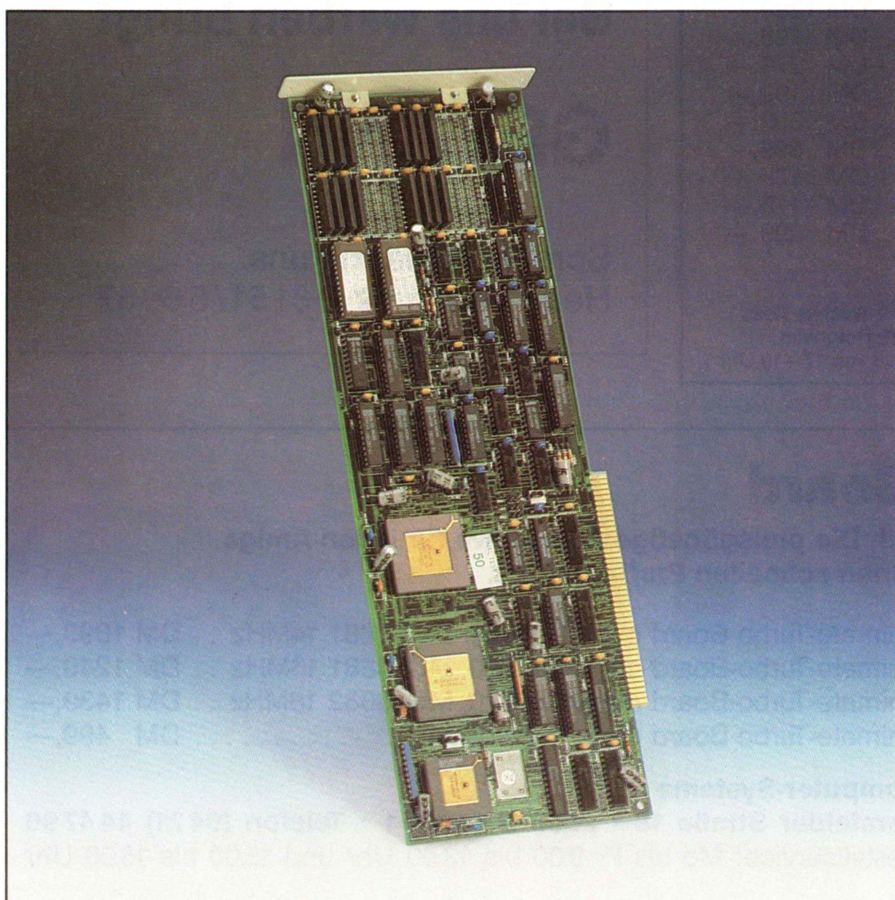
UNIX, Betriebssystem für Großrechner. Bestes Beispiel ist der Supercomputer Cray, der auch unter UNIX betrieben wird. Es sind viele unterschiedliche UNIX-Versionen im Umlauf. Commodore entschied sich für die Version V, Release 3.1. Da das Betriebssystem UNIX hauptsächlich auf Großrechnern zu Hause ist, liegt der Gedanke nicht weit, daß der 'kleine' AMIGA erheblich aufgerüstet werden muß. Hier sei nur erwähnt, daß UNIX über 300 Utilityprogramme beinhaltet. Diese im Arbeitsspeicher zu halten, würde wohl den vernünftigen Rahmen sprengen. Aus diesem Grunde werden mindesten 40 MByte freier Massenspeicher benötigt. Es gibt auch UNIX-Versionen, die mit nur einer Floppy-Diskette auskommen, ein Arbeiten mit dieser Version ist aber nicht sehr effektiv. Wenn das Betriebssystem läuft, wird weder AMIGA-DOS noch Exec benutzt. Daraus folgt, daß AMIGADOS nicht genutzt werden kann, wenn UNIX in Betrieb ist.

Zwei MByte FastMem und 500 KByte ChipMem sind mindestens vonnöten, vier MByte versprechen aber mehr Erfolg. Wie schon oben er-

wähnt, ist UNIX ein Multiuser-Betriebssystem, eine Weiterentwicklung des Multitasking-Betriebssystems. Bevor ein Benutzer unter UNIX arbeiten kann, muß er ein zuvor definiertes Paßwort eingeben. Unautorisierte Benutzer haben somit keinen Zutritt zum System. Mit der von Commodore vertriebenen seriellen Karte können sieben weitere Terminals installiert werden. Der AMIGA 500 ist dazu bestens geeignet.

Kommen wir zu der UNIX-Oberfläche. Commodore entwickelt extra für das Betriebssystem eine eigenständige, grafikorientierte Oberfläche. Leider ist sie zu keiner anderen kompatibel. Doch kommen wir zum Aufbau der Oberfläche. Die normale AMIGA-Maus ist das zentrale Eingabemedium, durch Drükken der linken Maustaste wird ein Menü aufgeklappt, in dem der Anwender die Aufgabe der Maus festlegen kann. Möchte er z.B. ein Fenster schließen, muß er das Feld mit einer abgebildeten Bombe anwählen, dann über dem gewünschten Fenster erneut die Maustaste betätigen, und es wird geschlossen. Andere Felder ermöglichen ein neues Fenster zu öffnen, zu vergrößern, zu verschieben und einiges mehr. Der Weg der Fensterbedienung scheint auf den ersten Blick etwas umständlich, beim Arbeiten geht es aber leicht von der Hand. Alle auf dem Bildschirm befindlichen Fenster besitzen einen Schatten, wobei ein räumlicher Eindruck entsteht. Zwei unterschiedliche Fenstertypen stehen zur Auswahl, ein Text- und ein Grafikfenster. Beide Fenstertypen unterstützen lediglich zwei Farben, jedoch bietet der Grafikspezialist AMIGA interessante Applikationen, die nur unter dem UNIX-AMIGA möglich sind.
Gadgets und Pop-Up-Menüs sind momentan in Planung. Wie das Handling und die grafikorientierte Oberfläche in der fertigen Version aussehen werden, war leider noch nicht in Erfahrung zu bringen, da noch fleißig programmiert wird.

Der AMIGA 2000 kann zu einer UNIX Maschine aufgerüstert werden. Im Bild sehen Sie die grafikorientierte Oberfläche, die z.Z. noch in der Entwicklung ist.



Der Grundbaustein für den UNIX-AMIGA ist die 68020-Karte mit 68881 FPU und 68851 MMU.

Kerneinheit des UNIX-AMIGA ist neben der 80 MByte-Festplatte und dem Streamer (für Festplattenbackup) die 68020-Karte, die mit dem Floating Point-Unit 68881 und der Memory-Management-Unit 68851 ausgerüstet ist. Optional kann auch die schnellere Unit 68882 eingesetzt werden, das wahlweise mit 20 MHz oder 25 MHz getaktet werden kann. Der 68020 wird mit 14.3 MHz getaktet, ebenso wie die FPU 68881. Wahlweise wird die Karte mit 2 oder 4 MByte 32-Bit-RAM ausgeliefert. Der Speicher ist hierbei autokonfigurierend. Die Karte ist voll kompatibel zum AMIGA-Betriebssystem und zu der meisten Software.

Transputer

Dieter Preiß, Chefingenieur von Commodore Braunschweig, hat einiges über das Commodore-Transputer-Konzept festgehalten. Wie Sie vielleicht wissen, bieten Transputer eine schnelle Abarbeitung von Programmen. Sie wird durch parallelen Transfer erreicht, die sogenannten Links. Das AMIGA-Transputerkit beinhaltet die Transputerkarte und das Betriebssystem HELIOS OS. Die Karte ist ausgerüstet mit einem T414 mit 15 MHz Taktfrequenz und 1 MByte RAM, das sich auf 4 MByte aufrüsten läßt. Die Schnittstelle zwischen AMIGA und Transputer erfolgt über einen AMIGA-100 Pin-Steckplatz. Der T414 kann leicht gegen einen T800 ausgetauscht werden, der eine erhebliche Geschwindigkeitssteigerung bringt. Durch Expansionskarten können bis zu 16 weitere Transputer installiert werden. Das Software-Kit beinhaltet HELIOS als Single User-Version und einen HELIOS C-Compiler. Das HELIOS Konzept ist sehr vielseitig und läuft sowohl unter einem Transputer als auch mit mehreren problemlos. Mit der Transputerkarte kann der AMIGA auch zu einem Netzwerk zusammengeschlossen werden. HELIOS OS verwaltet auch in diesem Fall alle Netzwerkstationen.

Fazit

Die Frankfurter Entwicklerkonferenz konnte mit Sicherheit einigen Entwicklern weiterhelfen. Die Atmosphäre der Konferenz war sehr locker und ungezwungen, hatten die Sprecher der einzelnen Themengebieten doch meistens einen Scherz auf Lager. Von der Fachpresse, wie der KICKSTART-Redaktion, konnten Beziehungen zu neuen Entwicklern im Bereich Hard- und Software geknüpft oder vertieft werden. Auch konnten etliche Informationen aus den Commodore-Entwicklern herausgelockt werden.

Die Planungen von Commodore sind vielversprechend, ob sie auch in die Tat umgesetzt werden, liegt schließlich und endlich an Commodore selbst. Aber auch die Fremdentwickler schlafen nicht, in der nahen Zukunft sind interessante Entwicklungen zu erwarten. Besonders der professionelle Bereich wird verstärkt berücksichtigt.

ENDE

Einladung

HANNOVER MESSE
CeBIT '89
Welt-Centrum Büro · Information · Telekommunikation
8. – 15. MÄRZ 1989

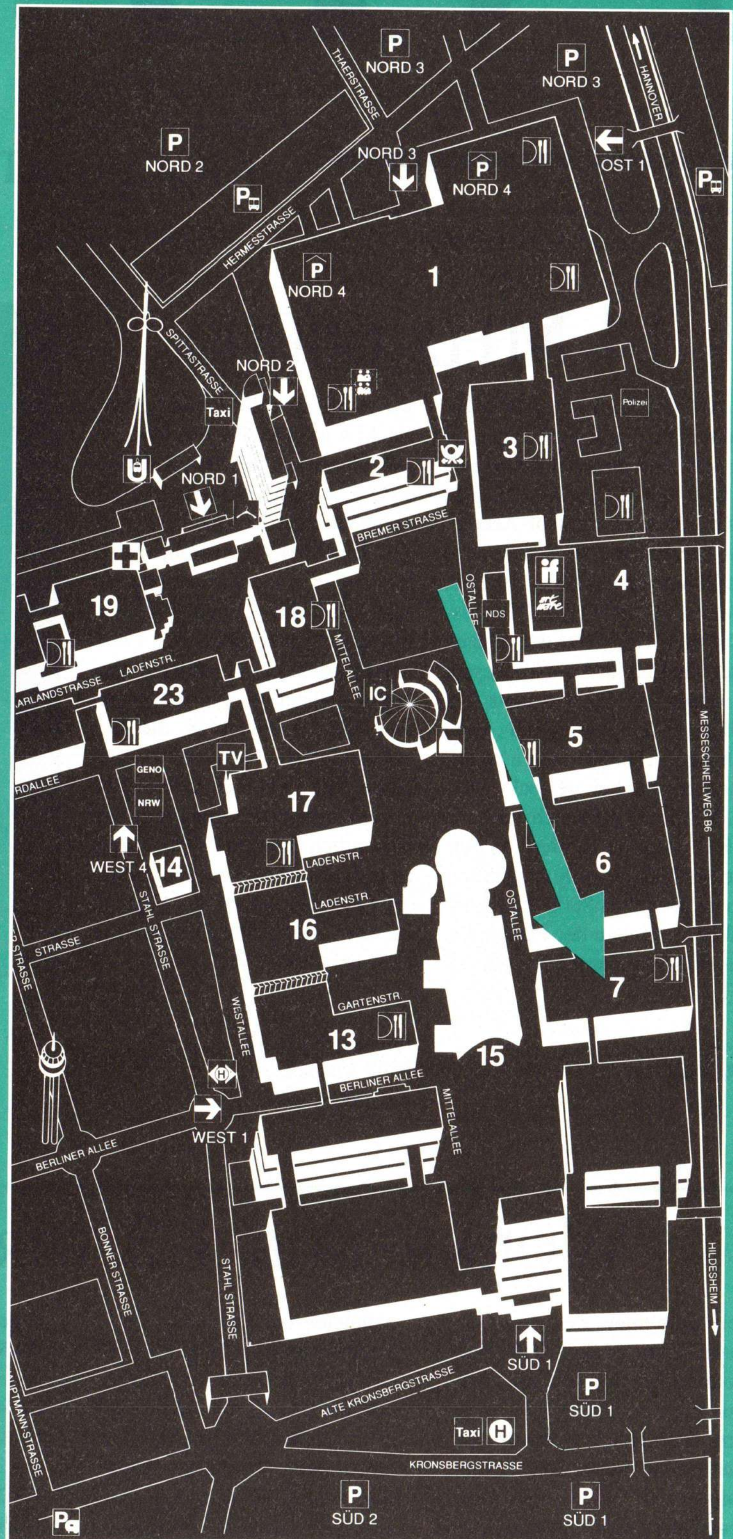
Halle 7, Stand E 42

Die KICKSTART-Redaktion lädt Sie herzlich zum Besuch unseres Standes auf der CeBIT '89 ein.

Dort können Sie direkt Fragen an uns stellen, Kritik, Anregungen oder Komplimente loswerden. Wir würden uns auch freuen, mit zukünftigen Autoren Kontakt aufzunehmen - kommen Sie doch einfach 'mal vorbei.

Weiterhin werden am Stand neue AMIGA-Produkte präsentiert, so z.B. das R.C.T., das neuartige Requester Construction Tool, mit dem man schnell und äußerst komfortabel eine professionelle Oberfläche erstellen kann. Oder der KICK-ASS, das leistungsstarke Entwicklungssystem durch den integrierten EDITOR, MONITOR, LINKER und DEBUGGER. Auch den KICK-ED können Sie sich näher anschauen, das enorm schnelle Scrolling wird Sie sicher begeistern. Einen Blick in die neuen AMIGA-Bücher können Sie ebenfalls werfen. Die AMIGA GFA-BASIC 3.0-Fibel, das AMIGA BASIC-Profibuch usw. liegen bereit. Selbstverständlich führen wir auch die Funktion des Junior-Prommers und der dazugehörigen Software vor.

Doch das ist noch nicht alles. Da unser Entwicklungslabor nicht untätig war, können wir Ihnen eine ganz besondere Überraschung bieten. Um was es sich dabei handelt, wollen wir noch nicht verraten, sonst wäre es ja keine Überraschung mehr. Soviel sei gesagt, es lohnt sich.



NEWS

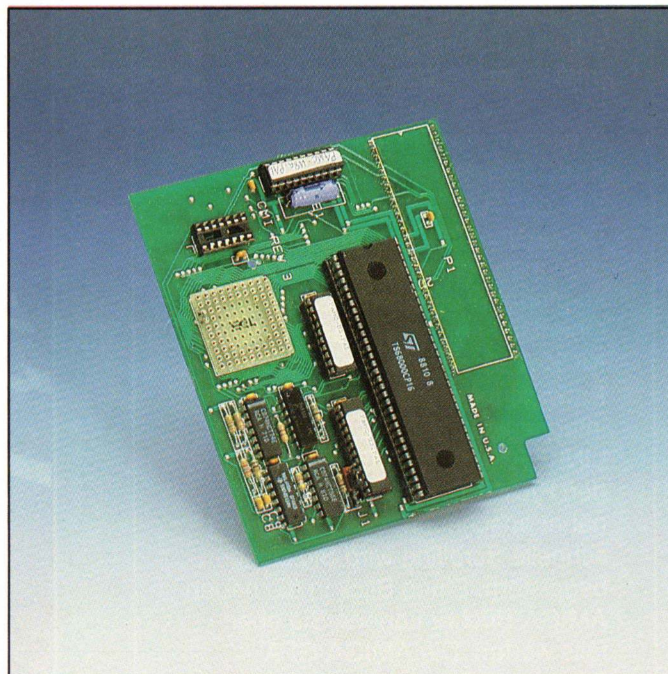
KICKSTART 3´89

14 MHz 68000

Der AMIGA wird mit 7.16 MHz getaktet, für viele Anwender zu langsam. Die amerikanische Firma CMI hat für diese AMIGA-Anwender den Processor-Accelerator entworfen. Der Accelerator besteht aus einer kleinen Platine, die einen 16 MHz 68000-Prozessor besitzt. Der Clou der ganzen Geschichte ist, daß der Prozessor mit doppelter Taktfrequenz betrieben wird. Zusätzlich kann noch die FPU 68881 in den bereitstehenden Sockel gesteckt werden. Mit der Workbench 1.3 kann die FPU genutzt werden, da eine neue Mathe-Library hinzugekommen ist. Der alte 68000-Prozessor muß beim Einbau der Platine aus seinem Sockel entfernt werden, wo stattdessen die Platine eingesteckt wird. Der Processor-Accelerator arbeitet mit allen AMIGA-Typen, da verschiedene Platinenabmessungen angeboten werden. Der AMIGA-Anwender sollte nicht zu viel erwarten, bedenkt man, daß die Customchips mit normaler Taktfrequenz (7.16 MHz) arbeiten. Ein kurzer Test ergab, daß der mit 14 MHz getaktete 68000er nicht an die Geschwindikeit von einer 68020-Karte mit FPU herankam. Eine kostengünstige Alternative zu den wesentlich teureren 68020-Karten ist mit dem Prozessor-Accelerator aber gegeben. Der Preis liegt bei ca. 500.- DM ohne 68881, mit FPU bei 1000.- DM.

IM
Borsigallee 18
6000 Frankurt
Tel. 068-410071

CSS
Auf der Warte 46
6367 Karben
Tel. 06039-5776



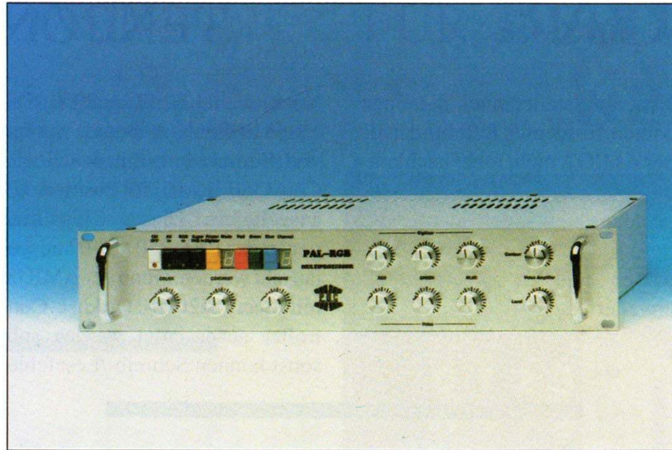
EIN MIT 16 MHZ GETAKTETER 68000-PROZESSOR BESCHLEUNIGT DEN RECHNER. ZUSÄTZLICH KANN DIE PLATINE MIT EINEM 68881 AUFGERÜSTET WERDEN.

WORKBENCH 1.3 WIRD AUSGELIEFERT

Seit geraumer Zeit im Gespräch, nun endlich lieferbar: Commodore beschert uns 1989 eine neue Kickstart 1.3 sowie eine Workbench 1.3 (mit den Versionsnummern 34.5 beziehungsweise 34.20). Für den deutschen Markt werden die Texte der neuen Produkte ausnahmslos ins Deutsche übersetzt. Kickstart 1.2 und 1.3 sind - laut Commodore - voll kompatibel, so daß man nicht gezwungen ist, seinen AMIGA umzurüsten. Die neue Kickstart ist nur nötig, wenn man eine autobootfähige Festplatte sein eigen nennt. Auch die neue resetfeste und bootfähige RAM-Disk RAD: benötigt die neue Betriebssystemversion. Commodore bietet für alle Umsteiger ein Update-Kit an, das die Workbenchversion 1.3D, die neue Extras-Diskette sowie ein deutsches Handbuch umfaßt. Es wird für ca. 70.- DM im Handel erhältlich sein. Wer seinen 1.2-AMIGA 500/2000 umrüsten will (die neuen AMIGAs werden schon mit Kickstart 1.3 ausgeliefert), kann sich die ROMs auf dem Ersatzteilweg besorgen, ein genauer Preis ist jedoch noch nicht bekannt.

PAL-RGB-Multiprozessor 2000

Die Firma Peter Biet Computerdesign (PBC) hat sich entschlossen, ihre gesamte Produktpalette neu zu überarbeiten. Herausgekommen ist dabei unter anderem ein PAL-RGB-Multiprozessor, der vor allem in der Videoverarbeitung einsetzbar ist. Im einzelnen enthält der Prozessor einen RGB-Farb-Splitter, einen Video-Color-Prozessor, einen Videoüberspielverstärker und anderes mehr. Der Prozessor soll in drei verschiedenen Versionen zum Preis zwischen 1098.-DM und 1598.-DM erhältlich sein.



PBC
Georg-Fischer-Str. 5
6415 Petersberg 2

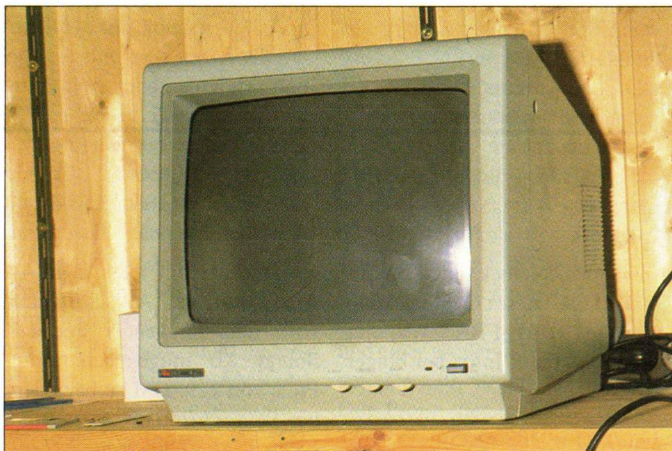
IEC-625-Bus

Weitere Aktivitäten (in Verbindung mit der Firma EDOTRONIK) bringen dem AMIGA-Anwender verschiedene neue Steckkarten für den AMIGA 2000. Da wäre zunächst ein IEC-625-Bus, eine Normschnittstelle, mit der man gleichzeitig bis zu 14 andere Geräte verbinden kann (auch andere Controller). Des weiteren gibt es noch ein Universal-Interface, mit dem es möglich ist, spätere Erweiterungen für seinen Rechner anzuschließen.

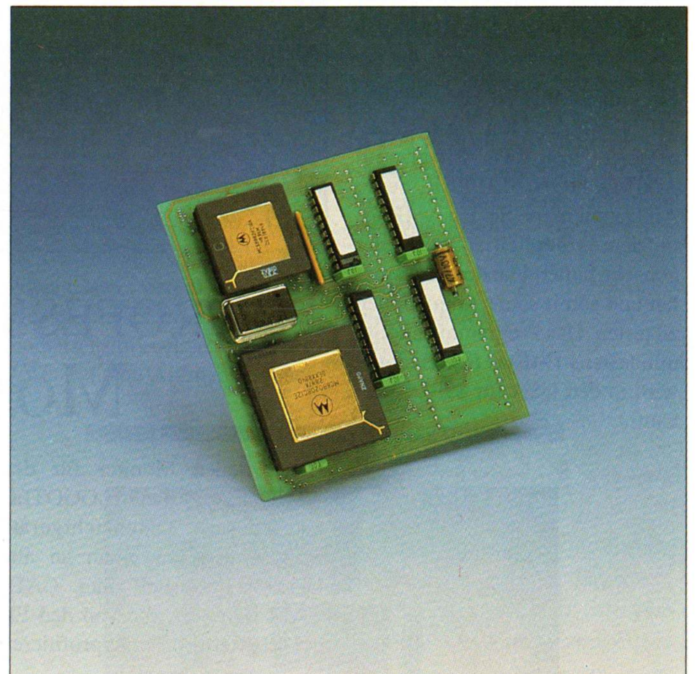
Stereo-Multisync-Monitor

Einen Multisync-Monitor, der einen kleinen 2 x 10 Watt Stereoverstärker beinhaltet, bietet die FIRMA CSS an. Normale Multisync-Monitore glänzen zwar mit einem guten Bild, mit einem Verstärker wurden die Geräte aber nicht ausgerüstet. Neuerdings wird der Stereomonitor auch mit zwei Breitbandlautsprechern ausgeliefert. Die maximale Auflösung des umgebauten Atec Multisyncs liegt bei 800 x 560 Pixeln. Die automatische, horizontale Synchronisation liegt zwischen 15.5 und 35 KHz, die vertikale zwischen 50 und 70 Hz. 0.31 mm beträgt der Dotabstand. Die Bildqualität ist ausgesprochen gut. Das Interlace-Flimmern wird gemindert.

CSS
Auf der Warte 46
6367 Karben 1
Tel. 06039-5776
Preis: 1400.- DM



Der Multisync-Monitor wird mit eingebautem Stereoverstärker und zwei Breitbandlautsprechern ausgeliefert.



Turbo Board

Die Firma CCS vertreibt eine 68020-Karte mit 68881 FPU, die mit 14 MHz Taktfrequenz arbeitet. Es kann aber auch der schnellere Bruder des Mathe-Coprozessors, der 68882 verwendet werden. Tests des Herstellers ergaben, daß mit dem 68882 Taktfrequenzen bis zu 30 MHz problemlos verkraftet wurden. Die kleine Platine findet im Sockel des 68000ers Platz. Ein Umschalten auf den 68000 Prozessor ist dadurch nicht möglich. Es ist kein 32-Bit breiter Bus von der Karte herausgeführt, ein Anschluß eines 32-Bit breiten RAM-Speichers erübrigt sich aus diesem Grunde. Der Preis des Boards liegt bei 1490.- DM mit FPU und 16 MHz Quarz. Auf Wunsch wird auch die schnellere FPU mit schnellerem Quarz ausgeliefert.

Neue Shell für den AMIGA

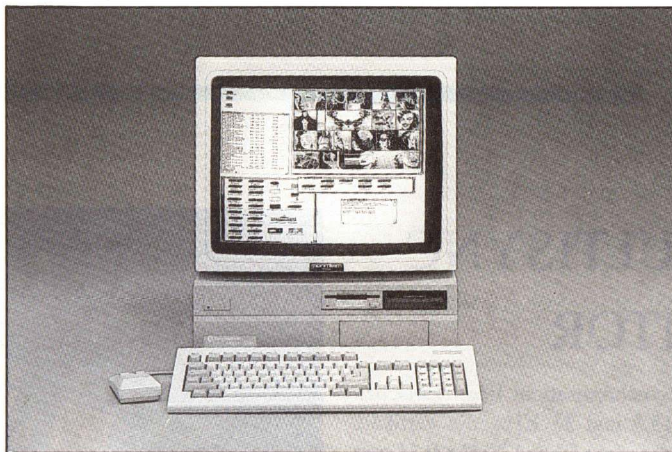
William S. Hawes, der Autor von Conman, bietet eine erweiterte Version seines Programms an. Sie macht das Arbeiten mit dem spartanischen CLI um vieles einfacher und komfortabler. So kann man, wenn man sich vertippt hat, den Befehl korrigieren oder alte Befehle noch einmal ausführen. Eine Neuerung stellen auch die eingebauten, residenten Befehle dar, die der AMIGA nicht mehr nachladen muß. WShell ist sehr flexibel gehalten, so daß der Benutzer das System seinen Anforderungen anpassen und es gegebenenfalls erweitern kann.

A2000A Festplattencontroller

Noch einmal AMIGA 2000: Die ersten Rechner (A-Board) wurden von Commodore mit dem PAL-Chip 380715-01 in Position U5 ausgeliefert. Dieser Chip muß gegen das PAL 380715-02 ausgetauscht werden, wenn der AMIGA mit einem A2090-Harddisk-Controller ausgestattet werden soll, sonst können Schreib-/Lesefehler auftreten. Da dies jedoch nur wenige Rechner betrifft und dieser Chip in den neueren AMIGA 2000B nicht mehr existiert, werden die AMIGA-Harddisk-Kits nunmehr ohne dieses PAL ausgeliefert. Man kann es aber dennoch als Ersatzteil bei Commodore beziehen.

Aktuelles für Animateure

Ein neues Mitglied der ANIMATION-Familie (von Hash Enterprises) ist das Programm ANIMATION:FLIPPER, eine Art Daumenkino für den AMIGA, mit dem es möglich ist, IFF-Bilder gepackt hintereinander abzuspielen und somit seine Animationen zu testen. Das Programm benötigt mindestens 1MByte Speicher und wird in den USA für $59.95 verkauft.



Professioneller Monitor

Einen neuen Monitor für den AMIGA 2000 bietet LOGOTEC an. Dieses "Datensichtgerät" wendet sich vor allem an alle Desktop-Publisher oder CAD/CAM-Benutzer, die von den Eigenschaften des Geräts profitieren sollen. Mit einer flackerfreien Höchstauflösung von 1008x800 Punkten in vier Graustufen (68Hz Bildwiederholung) ist der Viking 1 - so der Name des Monitors - bestimmt mal einen Blick wert. Informationen direkt bei:

LOGOTEC, Humboldtstraße, 2056 Glinde.

Computercorner in Essen

Am 5.März veranstalten der Computerclub Ruhrgebiet und das Jugendamt Essen einen Computercorner zum Thema Jugend und Computer (Ort: Jugendzentrum Essen, Papestraße). Geboten werden Diskussionsforen zu diesem Thema, Händlerstände und ein Flohmarkt. Computerclubs haben außerdem die Chance, sich an diesem Tage kostenlos vorzustellen. Wer Infos sucht oder sich anmelden will, schreibt an das Jugendwerk der Arbeiterwohlfahrt Essen, Pferdemarkt 7, 4300 Essen 1, oder ruft einfach mal an unter 0201/230949.

News in aller Kürze

*** Neue Druckertreiber mit einer Auflösung von 360x360 dpi bietet technicSupport *(Bundesallee 36-37, 1000 Berlin 31)* dem geneigten Anwender mit dem Schriftsatzprogramm AmigaTex an. Dieses Programm ist durch die hohe Druckqualität besonders für den wissenschaftlichen und schulischen Einsatz geeignet.

*** Merkens EDV *(Fuchstanzstraße 6A, 6231 Schwalbach, Tel. 06196/3026)* bietet in einer Sonderaktion das Einsteiger- und Anwender-Genlock miniGEN für 398DM an. Dieses Genlock kann auch als PAL-FBAS-Wandler genutzt werden. Infos bei obiger Adresse.

*** Zum Schluß noch etwas für Musiker: Es gibt einen neuen Stereo-Digitizer für alle AMIGA-Ausführungen. Als Software wird dem Anwender Perfect-Sound 2.0 mitgegeben. Zum Preis von 219.-DM zu beziehen bei *Casablanca, Nehrings kamp 9, 4630 Bochum 5 (Linden).*

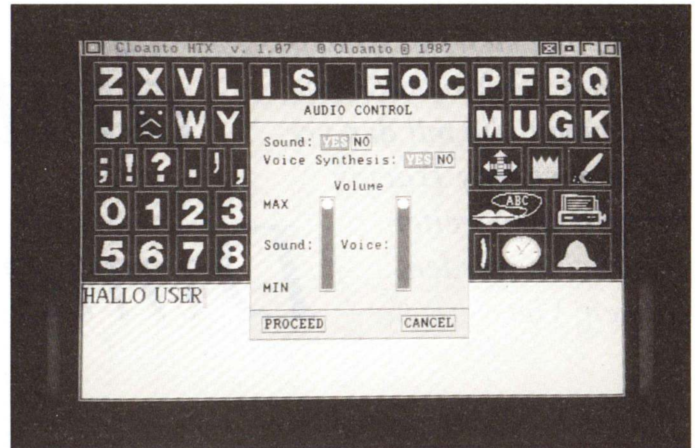
Sowjetische Mailboxen

Auf Einladung von sowjetischen Regierungstellen war eine Delegation der Deutschen Mailbox GmbH in Moskau, um eine Zusammenarbeit zu besprechen. Gesprächthemen waren der Betrieb des Mailbox-Systems und die Vermarktung von Mailboxdienstleistungen in der UdSSR. Vielleicht können Sie schon in naher Zukunft mit Mailboxen in Moskau, Leningrad und Kiew kommunizieren.

Fonts und Animationen aus Amerika

THE DISC COMPANY gibt ein Paket von Zeichensätzen - genannt "CaleFonts II" - heraus, das 17 Fonts in den Größen von 4 bis zu 64 Punkten umfaßt. Die ca. 149.-DM teure Sammlung ist vor allem für Anwender gedacht, die eine möglichst ansprechende Präsentation ihrer Produkte anstreben. Noch was von der DISC COMPANY: IMAGINE 4D (?) ist ein 3D-Grafik-Programm, das 3D-Animationen (PAL) in LoRes, HiRes und HAM unterstützt. Da hierbei das IFF-Format verwendet wurde, können selbsterstellte Bewegungsabläufe in anderen Programmen weiterverwendet werden. Ein Preis ist noch nicht bekannt.

Textverarbeitung für Behinderte und Kinder



Cloanto-HTX nennt sich eine Textverarbeitung, die speziell für Behinderte und Kinder entwickelt wurde. Ein Modus für Blinde ist ebenfalls integriert. Der Bildschirm teilt sich in zwei große Felder, das obere beinhaltet das Eingabemedium mit der Maus, das untere ein normales Textfeld. Die Bedienung ist speziell den Bedürfnissen von Behinderten angepaßt. Beispielsweise kann ein Blindenschriftdrucker angesteuert werden. Die Aufmachung des Programms läßt das Programm aber auch für Kinder interessant werden. Z.Z. ist es nur in englischer bzw. italienischer Sprache erhältlich, eine deutsche Version ist aber in Planung. Cloanto-HTX ist besonders interessant für Behindertenschulen, Krankenhäuser oder Pflegeheime.

Cloanto
Via G.B. Bison, 24
33100 Udine
Italien

AMIGA-Tower

Was IBM kann, können wir schon lange, dachte sich ein AMIGA-Entwickler, und entwarf den AMIGA-Tower. Dieser wird, wie bei 'Computertürmen' so üblich, nicht auf, sondern neben dem Schreibtisch plaziert. Auf dem Schreibtisch finden sich lediglich die Tastatur und der Monitor. Ein Tower-Gehäuse suggeriert dem Computeranwender mehr Leistung des Rechners, nicht anders verhält es sich mit dem AMIGA-Tower. Grundbaustein bleibt freilich das Motherboard des A2000B. Der Standard-Tower besitzt des weiteren 2 x 3.5" Laufwerke, 3 MByte RAM, Kickstart 1.3 mit integriertem Virusfinder (direkt beim Booten wird ein Virus erkannt), autobootende 20 MByte-Festplatte und 68000-Prozessor (16 MHz) mit Coprozessor 68881. Der Preis des Standart-Towers liegt bei 7950.- DM. Außerdem wird der AMIGA-Tower auch mit einem 68020-Prozessor und einer größeren Festplatte ausgeliefert (30, 40 und 60 MByte). Die Preise liegen beim Turbo-Tower zwischen 8850.- DM und 9390.- DM.

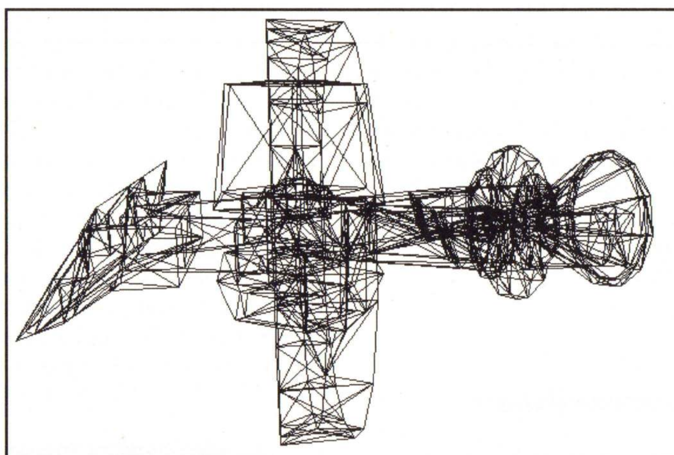
VON ANDREAS KRÄMER

Objekte der dritten Dimension

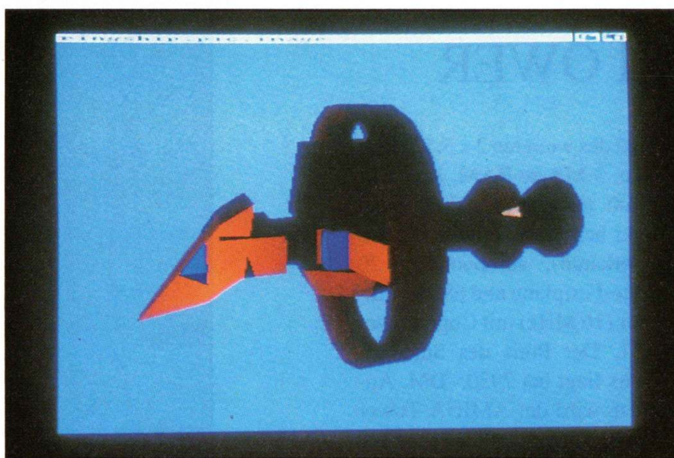
Das Softwarehaus ANTIC hat sich der Erstellung von 3D-Modellen angenommen, die mit den verschiedenen Ray Tracing Programmen weiterverarbeitet werden können. Vier verschiedene Disketten können erworben werden. Die Disketten sind hierbei in unterschiedliche Themen gegliedert:Microrobot, Future, Human und Architectural.

Die Programme, für die die 3D-Modelle angeboten werden, sind Sculpt-3D, Sculpt-Animate 3D, Modeler-3D und Videoscape 3D, wobei die Modelle beispielsweise auch von Turbo SILVER genutzt werden können, nachdem sie mit dem Programm Interchange konvertiert würden. Auf jeder Diskette befinden sich immer mehrere verschiedene Modelle, die mitunter sehr komplex gehalten sind. Nach dem Einladen, beispielsweise mit Sculpt-3D, dauert der Aufbau des Modells schon einige Sekunden, auch ein 68020-Prozessor schafft hierbei wenig Abhilfe. Setzen von Blickwinkel, Verschieben des Modells usw. verlangen deshalb viel Feingefühl und Zeit.

Die Modelle aller Disketten sind, wie schon erwähnt, sehr komplex, dementsprechend dauert auch die Berechnung der 'getracten' Modelle ausgesprochen lange. Das Ergebnis entschädigt aber den Grafikbegeisterten um einiges. Um den Umfang eines Sculpt-3D Modells zu verdeutlichen, ein paar nähere Angaben: Einige Modelle bestehen aus mehreren Dateien, so z.B. der Roboter auf der Diskette Microrobot.



Das Drahtgittermodel einer Sphinx, eingeladen mit Sculpt-3D.



Das Weltraumschiff nach langer Berechnungszeit.

Er umfaßt 39 Objekte, 3052 Vertices und 5710 Faces. Diese Angaben sind Durchschnittswerte, kleinere bzw. größere sind ebenfalls vorhanden. Wer sich mit Ray Tracing-Programmen auskennt, weiß, wieviel Zeit und Arbeit das Erstellen von derart komplexen Modellen in Anspruch nimmt.

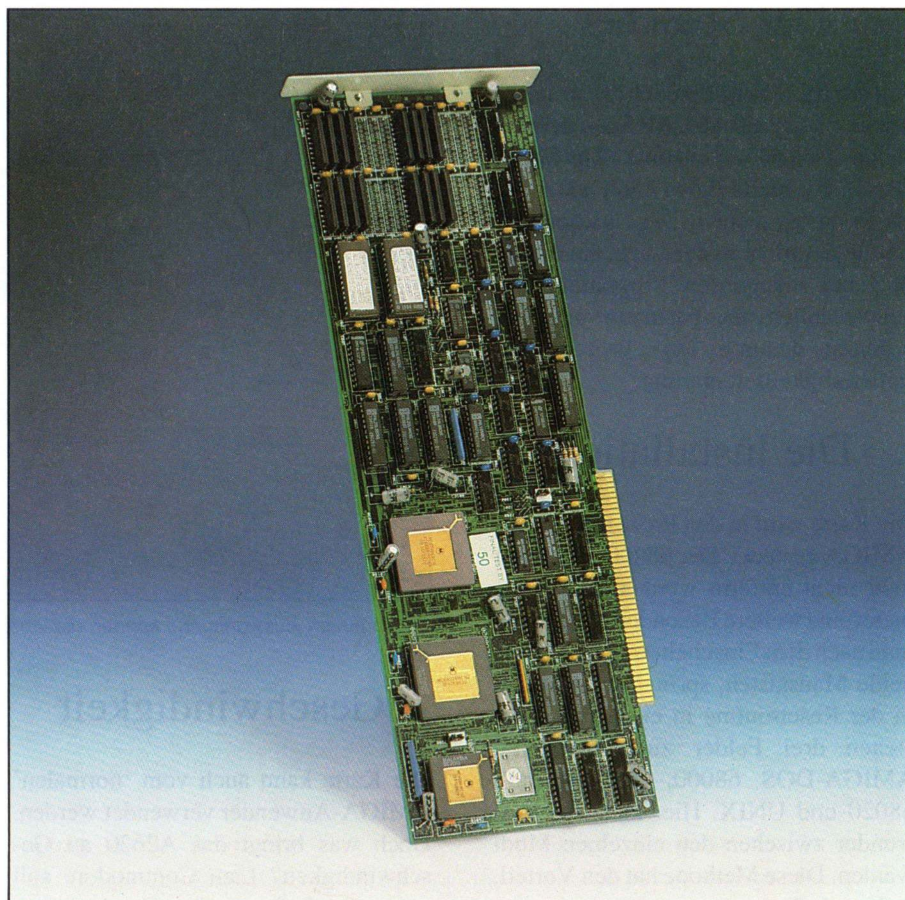
Jeder Modellsammlung liegt ein Blueprint der verschiedenen Modelle bei, der Anwender kann sich dadurch ein schnelles Bild über deren Struktur verschaffen. Den Nutzen der Modelle muß jeder Anwender für sich selber entscheiden, begeisterte Anwender in Sachen Grafik und Ray Tracing können sich die Sammlungen ruhig näher anschauen. Der Preis liegt mit knapp 90.- DM allerdings recht hoch.

Anbieter:
IM
Borsigallee 18
6000 Frankfurt

Preis: 89.- DM

ENDE

68020-Prozessoren sind abwärtskompatibel, das bedeutet, daß der 68000-Prozessor gegen einen 68020 getauscht werden kann. Der Vorteil des 68020-Prozessors besteht darin, daß er wesentlich schneller ist als sein kleiner Bruder, und daß er über einen 32 Bit breiten Bus verfügt. 68020-Prozessorkarten gibt es mittlerweile etliche, die von Commodore vertriebene A2620-Karte besitzt aber einige Besonderheiten.



VON ANDREAS KRÄMER

68020-PROZESSOR-KARTE

Hauptsächlich ist die A2620 für den Einsatz unter UNIX gedacht, dafür spricht auch die 68851 MMU, die für die Speicherverwaltung zuständig ist. Keine andere Karte verfügt über diesen Motorola-Chip. Der Nachteil: Da der Chip in keiner außerordentlich großen Stückzahl gefertigt wird, ist er recht teuer. Der Preis der Karte steigt dementsprechend. Kompatibel zum AMIGA-DOS-Betriebssystem bleibt die Karte trotzdem.

Der mathematische Coprozessor

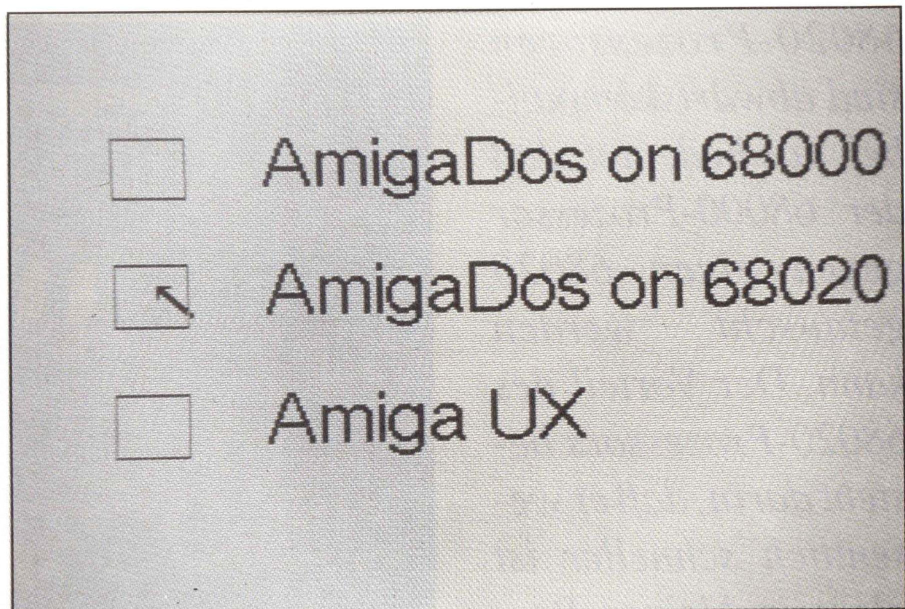
Neben dem 68020-Prozessor und der MMU findet sich natürlich auch das Floating Point Unit (FPU) 68881. Der mathematische Coprozessor 68881 kann auch gegen seinen schnelleren Bruder, den 68882, ausgetauscht werden. Anwendungen, die den Coprozessor nutzen, erhalten noch einmal eine erhebliche Geschwindigkeitssteigerung. Der 68881 wird mit 14 MHz getaktet, ebenso wie der 68020, optional kann auch mit 20 bzw. 25 MHz getaktet werden. Der schnellere Quarz wird am dafür vorgesehenen Platz auf der Platine plaziert; bei unserer Testkarte mußte er angelötet werden, ein Sockel wird aber hoffentlich bei den käuflichen Versionen vorhanden sein.

Der Speicher

Auf der Karte befinden sich des weiteren 2 bzw. 4 MByte RAM. Als Speicherbausteine dienen sogenannte Zip-Chips. Das 32 Bit breite RAM kann leider nur als Fastmem dienen. Die Customchips sind leider nicht in der Lage, auf diesen Speicher zuzugreifen. Programme, die hauptsächlich im Fastmem arbeiten, erfahren dadurch eine weitere Geschwindigkeitssteigerung.

Die Installation

Die Karte wird in den Prozessorslot des AMIGA gesteckt. Der 68000-Prozessor muß nicht entfernt werden. Die Karte bietet eine weitere Besonderheit. Drückt man nach dem Einschalten des Rechners beide Maustasten, springt der AMIGA in der Resetroutine in ein Menü. Hier stehen drei Felder zur Verfügung: AMIGA-DOS 68000, AMIGA-DOS 68020 und UNIX. Hier kann der Anwender zwischen den einzelnen Modi wählen. Diese Methode hat den Vorteil, daß auch Programme gestartet werden können, die nicht mit dem 68020-Prozessor arbeiten. Der letzte Menüpunkt installiert das UNIX-Betriebssystem, das z.Z. noch entwickelt wird.



In einem Auswahlmenü können die verschiedenen Modi des AMIGA installiert werden

Geschwindigkeit

Die Karte kann auch vom 'normalen' AMIGA-Anwender verwendet werden. Doch was bringt das A2620 an Geschwindigkeit? Laut Commodore soll angepaßte Software eine Geschwindigkeitssteigerung von bis zu 400% erhalten. Auf die genaue Definition der verschiedenen Benchmarks möchte ich verzichten, stattdessen gebe ich Ihnen deren Berechnungszeiten im Vergleich zum 68000-Prozessor. Die genauen Zeiten können Sie in der nebenstehenden Tabelle ablesen.

Fazit

Die A2620-Karte bietet einiges, besonders für den, der gedenkt, seinen AMIGA 2000 früher oder später mit dem Betriebssystem aufzurüsten. Mit ca. 6500.- DM (mit 2 MByte RAM) ist die Karte nicht gerade billig und wird wohl nur für betuchtere AMIGA-Anwender von Interesse sein.

Anbieter:

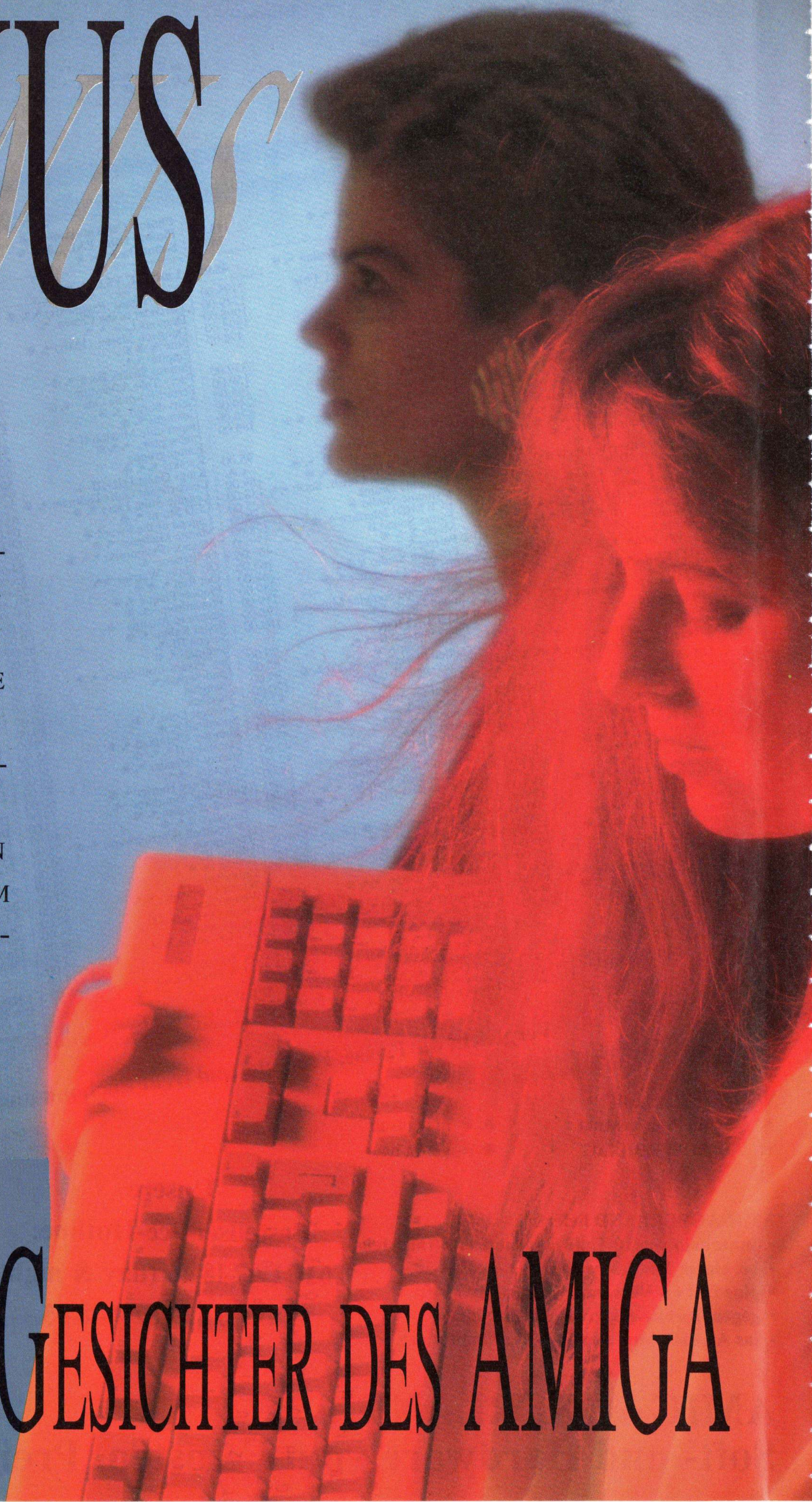
CSS
Auf der Warte 46
6367 Karben 1
Tel. 06039/5776

```
Der 68000 Prozessor erreicht folgende Zeiten:

Whetstone C Benchmark V1.0

End Module 1.    Benchtime(sec) = 0.000000
End Module 2.    Benchtime(sec) = 1.640000
End Module 3.    Benchtime(sec) = 12.139999
End Module 4.    Benchtime(sec) = 1.340000
End Module 5.    Benchtime(sec) = 0.000000
End Module 6.    Benchtime(sec) = 11.419999
End Module 7.    Benchtime(sec) = 26.500000
End Module 8.    Benchtime(sec) = 55.019998
End Module 9.    Benchtime(sec) = 4.320000
End Module 10.   Benchtime(sec) = 0.000000
End Module 11.   Benchtime(sec) = 20.720000
End Module 3.    Benchtime(sec) = 7.440000
End Module 8.    Benchtime(sec) = 52.560000
End Module 9.    Benchtime(sec) = 3.780000

Standard Whetstone Result
Benchtime(sec) = 133.100008
KWhets/sec     = 75
Result with subroutine call replaced with subroutine
code
Benchtime(sec) = 125.399994
KWhets/sec     = 79
```

```
Die Benchmarks mit dem 68020 Prozessor und der 68881
FPU:

Whetstone C Benchmark V1.0

End Module 1.   Benchtime(sec) = 0.000000
End Module 2.   Benchtime(sec) = 0.220000
End Module 3.   Benchtime(sec) = 1.700000
End Module 4.   Benchtime(sec) = 0.300000
End Module 5.   Benchtime(sec) = 0.000000
End Module 6.   Benchtime(sec) = 1.460000
End Module 7.   Benchtime(sec) = 1.540000
End Module 8.   Benchtime(sec) = 11.480000
End Module 9.   Benchtime(sec) = 3.060000
End Module 10.  Benchtime(sec) = 0.000000
End Module 11.  Benchtime(sec) = 1.020000
End Module 3.   Benchtime(sec) = 1.080000
End Module 8.   Benchtime(sec) = 8.220000
End Module 9.   Benchtime(sec) = 2.880000

Standard Whetstone Result
Benchtime(sec) = 20.780000
KWhets/sec     = 481
Result with subroutine call replaced with subroutine
code
Benchtime(sec) = 16.720000
KWhets/sec     = 598
```

VON CHRISTIAN KELLER

JANUS

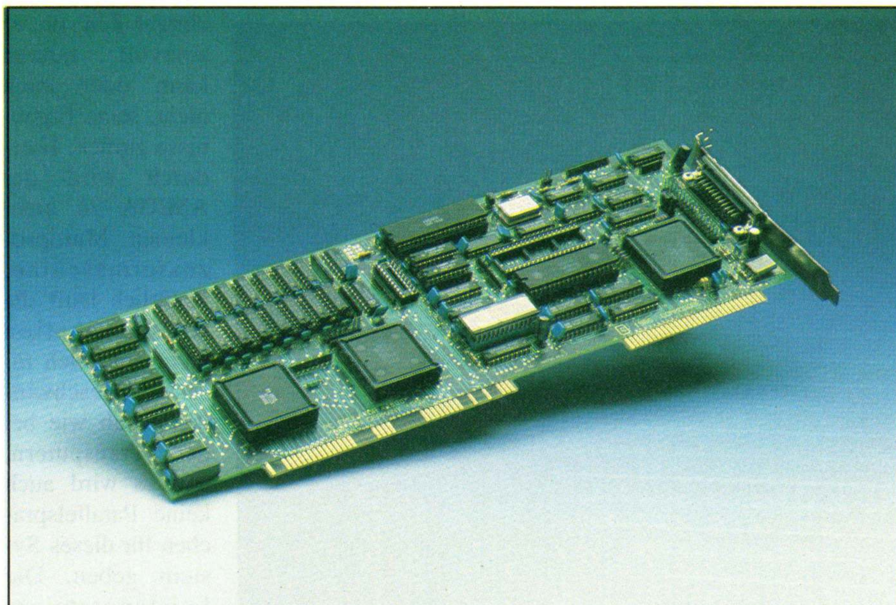


COMMODORE WÄHLT FÜR SEINE IBM-KOMPATIBLEN STECKKARTEN DEN NAMEN BRIDGEBOARD. DIESE BRÜCKEN, MITTLERWEILE IN VERSCHIEDENEN VERSIONEN ERHÄLTLICH, VERBINDEN DEN AMIGA MIT DEM SOFTWARE-SCHLARAFFENLAND DES MS-DOS.

DIE ZWEI GESICHTER DES AMIGA

Mittlerweile gibt es die PC-Emulatoren (wenn man hier überhaupt noch von Emulation reden kann) bereits in der dritten Generation. Emulieren ist vielleicht der falsche Ausdruck, da der PC ja zu 90 Prozent real existiert.

Urvater der Familie war das Sidecar, ein externes Zusatzteil für den A1000. Im Gegensatz zur AT-Karte, die nun langsam aber sicher auf den Markt kommt, war der Abstand zwischen der ersten Vorstellung und der Auslieferung recht gering. Vorgestellt auf der COMDEX 1986, war sie Ende desselben Jahres auch schon zu kaufen. Technisch gesehen war es ein abgespeckter Commodore PC10, der die Interface-Hardware aufgepflanzt bekam.



Das A2088 Bridgeboard - der Nachfolger des legendären Sidecar

Nachfolger des Sidecars war das A2088 Bridgeboard, das für den A2000 konzipiert war. Die Leistungsmerkmale entsprechen denen des Sidecars; durch Verwendung von hochintergierten Bausteinen war es aber möglich geworden, den kompletten PC-Clone auf einer einzigen Steckkarte unterzubringen.

Jüngster Sproß der Familie ist das A2286, die AT-Karte. Im Gegensatz zu seinen Eltern ist der Kleine ein Kraftprotz geworden und imponiert durch seine Leistungsfähigkeit. Durch die verspätete Geburt ist die AT-Karte aber schon etwas von der Entwicklung überholt worden.

Was ist das Besondere am PC im AMIGA?

Mancher wird sich fragen, was für Vorteile er hat, wenn er sich ein Bridgeboard, anstatt einen zweiten Rechner kauft. Zuerst fallen einmal die Vorteile beim Platzbedarf und bei den Anschaffungskosten ins Auge. Man benötigt nur ein Gehäuse auf dem Schreibtisch, nur einen Monitor und eine Tastatur. Speziell beim A2088 fällt zudem der finanzielle Vorteil ins Gewicht.

Wesentlich interessanter ist die Kopplung der beiden Rechner. Durch diese sehr innige Umarmung der beiden Systeme ist es möglich, weitestgehend auf die Resources des anderen Rechners zurückzugreifen - angefangen bei der gemeinsamen Benutzung von Peripheriegeräten wie Drucker oder HardDisk bis hin zum Auslagern von Rechenprozeduren auf das andere System, das die gestellte Aufgabe besser lösen kann (weil es zum Beispiel eine FPU implementiert hat.).

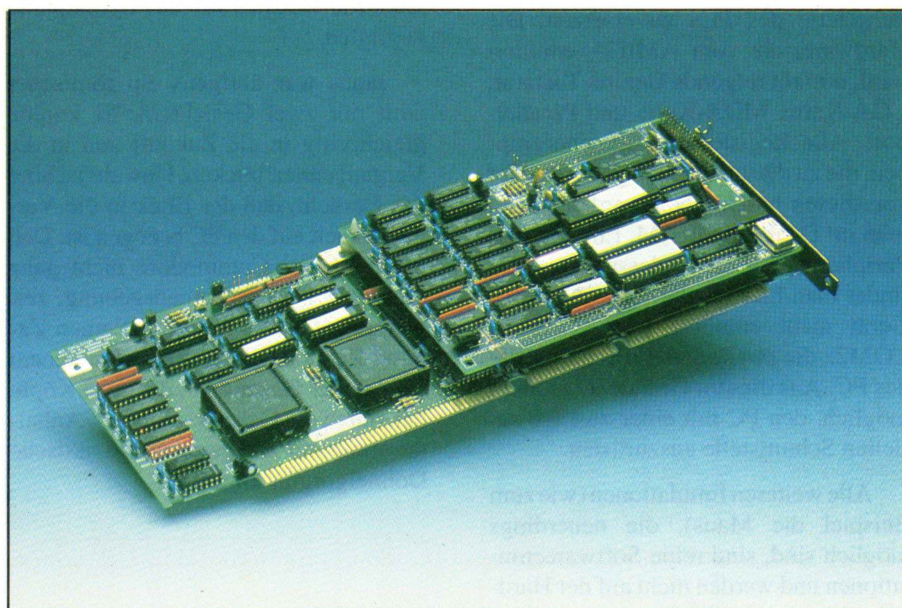
Damit diese Zusammenarbeit vonstatten gehen kann, brauchen die Rechner natürlich eine Verbindung zwischen den beiden Systemen. Dieser Kommunikationspfad ist das DualPortedRam. Dieser Speicherbereich, 128 kByte groß, kann von beiden Prozessoren benutzt werden. Hier findet jegliche Kommunikation und jeder Datenaustausch statt. Aufgeteilt ist das DPR in folgende Bereiche:

- 64 kByte Pufferspeicher. Hier werden Daten ausgetauscht. Ebenfalls in diesem Bereich angesiedelt ist der Janus-Handler, der die Verwaltung des Ganzen übernimmt.

- 16 kByte Parameterspeicher. Dieser dient zur Ablage aller Strukturen und Variablen, die von der Betriebssystemsoftware benötigt werden.

- 32 kByte dienen als Speicher des ColorGraphicsAdapters, der vom AMIGA emuliert wird.

- 8 kByte werden für die Emulation des



Der jüngste Spross der DOS-Familie - die A2286 AT-Karte

MonochromDisplayAdapters verwendet.

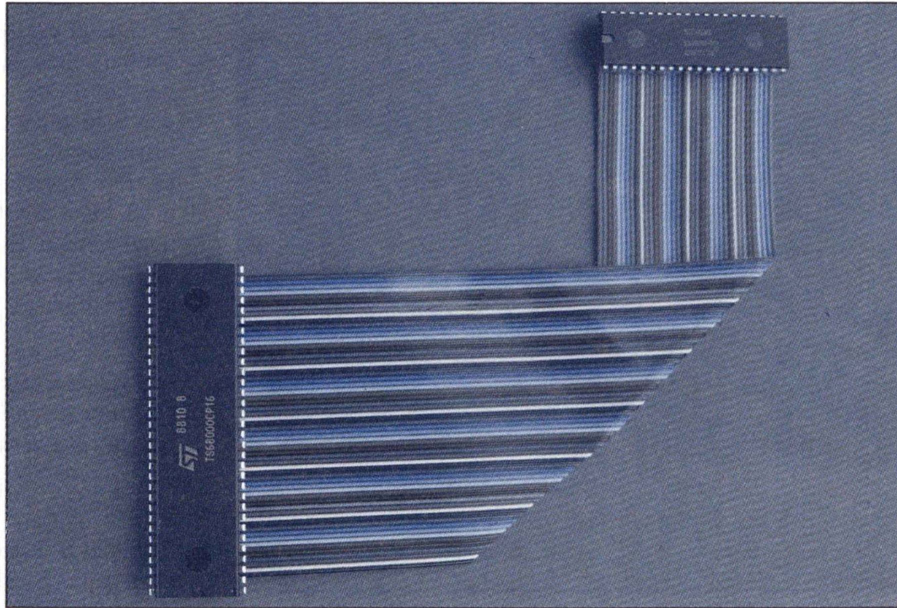
- 8 kByte emulieren diverse I/O-Register, die zur emulierten Hardware gehören.

Die acht kByte I/O-Speicher werden teilweise durch Hardware unterstützt, da manche Register sehr schnelle Reaktionen ausführen müssen, wozu der AMIGA nicht in der Lage ist. Diese 128 kByte, die im AMIGA-Adreßbereich in einem Stück verwaltet werden, liegen im PC-Speicher bunt gemischt an den verschiedensten Stellen. Die Videospeicher sind wegen der normalen Lage der Videokarten auf B0000h und B8000h fixiert. Das Parameter-und Puffer-RAM findet man je nach Rechnertyp an verschiedenen Stellen (z.B. Puffer-RAM bei E0000h für Sidecar und PC-Karte oder D0000h für die AT-Karte). Lediglich die 8 kByte I/O-RAM findet man nicht im normalen Adreßbereich des PCs, sondern im I/O-Adreßraum. Hier nehmen sie nur ein Kilobyte Raum in Anspruch, der Großteil der Adressen ist nicht benutzt.

Mit dieser Ausstattung ist es dem AMIGA möglich, die nicht vorhandene Peripherie des PCs zu ersetzen. Die Hardware, die vom AMIGA emuliert wird, umfaßt folgende Geräte: Tastatur, CGA-Karte, MDA-Karte und Parallelport. Alle Register und Speichergruppen, die zu diesen Geräten gehören oder von diesen beschrieben werden, findet man im DualPortedRAM. Ein weiteres Peripheriegerät bzw. dessen Register findet man hier wieder, obwohl dieses Gerät nicht emuliert wird. Dies ist COM2:, die zweite serielle Schnittstelle des PC. Aus diesem Grunde ist es nicht möglich, den PC mit einer zweiten seriellen Schnittstelle auszurüsten.

Alle weiteren Emulationen (wie zum Beispiel die Maus), die neuerdings möglich sind, sind reine Softwareemulationen und werden nicht auf der Hardwareebene nachgebildet.

Die Kommunikation der beiden Systeme wird vom Janus-Handler auf der PC-Seite und von der Janus-Library im AMIGA überwacht. Die Library wird beim Starten des AMIGA in das Betriebssystem eingebunden. Von dieser wird dann der Janus-Handler in das DPR übertragen. Dieser Handler wird vom PC als BIOS-ROM-Erweiterung identifiziert und initialisiert. Von diesem Zeitpunkt an laufen einige Betriebssystemroutinen (Interrupt 13H, der für die Handhabung der Massenspeicher zuständig ist) durch diesen Bereich, was sie für den AMIGA zugänglich macht. Zudem wird dem AMIGA die Möglichkeit gegeben, selbst Interrupts auf dem PC auszulösen. Der Effekt: Der AMIGA kann zum Beispiel auf die PC-Festplatte zugreifen.



Durch die Kommunikationssoftware ist die Grundlage zur Zusammenarbeit der verschiedenen Prozessoren geschaffen.

Janus war übrigens ein römischer Gott mit zwei Gesichtern. Er konnte gleichzeitig in die Zukunft und in die Vergangenheit blicken. Unwahr ist aber das Gerücht, daß der Blick in die Vergangenheit auf den PC bezogen ist. Daß man sich bei Commodore nicht ganz einig war bei der Namensgebung, zeigen einige Tasks mit dem Namen Zaphod. Zaphod B. ist eine Figur aus einem Roman der Gegenwart, die zwei Köpfe hat, durch die Galaxis schippert, Anhalter mitnimmt und gerne pangalaktische Donnergurgler drinkt.

Services

Mit der Version 2.0 wurde eine neue Kommunikationsmöglichkeit der beiden System geschaffen. Diese Services genannten Programme bieten die Möglichkeit, die beiden Prozessoren noch enger zusammenarbeiten zu lassen. Ein Service stellt, stark vereinfacht, einfach ein Unterprogramm dar, das auf der anderen CPU abläuft. Der aufrufende Rechner übergibt dem Service beim Aufruf seine Parameter und erhält nach einiger Zeit, die er sinnvoll nutzen kann oder auch nicht, seine Ergebnisse zurück. Hierdurch wird der AMIGA zu einer kleinen Multiprozessormaschine. Natürlich läuft die Kommunikation nicht mit solch rasender Geschwindigkeit ab wie bei den Transputern, und es wird auch keine Parallelsprachen für dieses System geben. Die Leistungssteigerung des Systems ist trotzdem erheblich. Die Fähigkeiten des Services gehen aber noch über die einfache Auslagerung von Prozeduren hinaus. Es ist durchaus denkbar, mit einem Service einen Grafikbildschirm für den PC zu emulieren, der die Auflösung des AMIGAs voll ausnutzt (Flimmerlace eingeschlossen). Die Software, die diesen Bildschirm ausnutzt, müßte dann natürlich auch speziell für die Bridgeboards geschrieben werden. Oder wie wäre es mit einem eigenen Druckerdevice mit großem Puffer, das der AMIGA dem PC zur Verfügung stellt, einer Mathematik-Library für den AMIGA, die den Coprozessor des PCs benutzt? Die Möglichkeiten sind fast unbegrenzt.

Soweit also die Erläuterung der grundlegenden Mechanismen, die zur Kommunkation PC-AMIGA zur Verfügung stehen. Darauf aufbauend sind alle Dienstprogramme programmiert, die zum Bridgeboard dazugehören. Die Features dieser Software sind an anderer Stelle in dieser Ausgabe erläutert.

Christian Keller

Alles drin - Alles dran

Die neue Janus-Software

Die neue Software, die mit den Brückenkarten geliefert wird, ist insgesamt erweitert und verbessert worden. Neben diversen Änderungen an den schon bekannten Programmen wurden mehrere Programme hinzugefügt, die einiges Nützliches anzubieten haben. In der jetzt verfügbaren Version ist alles Wichtige und Brauchbare implementiert ist. Die lange Wartezeit scheint sich also gelohnt zu haben.

Am erfreulichsten ist die Tatsache, daß alle PC-Zusätze, die für die verschiedenen AMIGA-Modelle angeboten wurden oder noch werden, mit der Software zusammenarbeiten. Sidecar, PC-Karte oder AT-Board, alle können die Software benutzen. Im letzten Heft war zwar eine Notiz, daß es Probleme mit der Sidecar geben würde, dies hat sich aber mittlerweile geklärt. Der Grund für die Fehlfunktion war eine Fehlkonfiguration des Datentransfersegmentes, die von der Standardeinstellung der Software herrührte.

Installation

Die Installation der neuen Software ist dank eines Installationsprogrammes sehr komfortabel. Genaugenommen sind es mehrere Installationsprogramme, die auf die verschiedenen Hardwaregegebenheiten abgestimmt sind. So gibt es ein Installations-Utility für das Sidecar, das den anderen LPT1-Treiber, der vom Sidecar benötigt wird, berücksichtigt. Ein anderes Programm ist darauf eingerichtet, nur die Device-Treiber und Librarys inklusive der von ihnen benötigten Programme einzurichten. Die Installation wird über ein Icon gestartet. Man erhält eine Liste auf dem Bildschirm, welche Dateien auf die Zieldiskette oder Platte kopiert werden sollen. Von dieser Liste können Dateien deselektiert werden. Daraufhin wird das Ziel-Device überprüft. Wiederum erscheint eine Liste, diesmal jedoch von allen Dateien des Ziellaufwerkes. In einer Anzeige am oberen Rand wird die Zahl der Blöcke mitgeteilt, die freigemacht werden müssen, um alle PC-Dateien kopieren zu können. Es werden auch einige Dateien markiert, die nicht zwingend notwendig zur Funktion des Rechners gebraucht werden. Diese Liste kann man verändern und so den benötigten Platz freigeben. Das Installationsprogramm löscht dann die markierten Dateien und überträgt die PC-Software auf die Zieldiskette. Als letzter Schritt, der aber von Hand erfolgen muß, ist die korrekte Einstellung des Puffersegments zu überprüfen. Dieses muß bei der Sidecar und dem A2088 auf E000 eingestellt werden, beim AT-Clone auf D000. Die Einstellung erfolgt im Programm PCPrefs. Hier lag auch der Grund für die anfängliche Fehlfunktion meiner Sidecar. Steht der Segmentwähler falsch, findet man zwar auf der PC-Seite die Daten an der richtigen Stelle (z.B. mit einem Speichermonitor), der AMIGA verwaltet diese aber nicht korrekt. In der Anleitung heißt es aber optimistisch, wenn der PC den Janus-Handler finden würde, wäre alles in Butter.

Hat man die Installation beendet, kann das System mit der erstellten Diskette gebootet werden. Was einem das Installationsprogramm natürlich nicht abnehmen kann, ist die Änderung der Startup-Sequence. Da die PC-Software einige Utilities enthält, die am besten beim Systemstart aufgerufen werden, sollte man diese in die Startup-Sequence einbauen. Dies wären zum Beispiel *AMouse*, *PCDisk* oder *TimeServ*.

Aller Anfang ist schwer, so auch bei der Installation eines neuen Gerätes. Hat man dies aber hinter sich, steht dem Arbeiten mit dem Bridgeboard nichts mehr im Weg.

Fensterln

Am Programm *PCWindow* wurden einige Modifikationen durchgeführt. Als neuer Menüpunkt ist "*Adjust Key Timing*" hinzugekommen. Hier kann der Zeitraum zwischen der Übertragung zweier Tastencodes seingestellt werden. Probleme mit manchen Programmen, die sehr hardwarenah auf die Tastatur zugreifen, können so eliminiert werden.

```
ENTER NUMBER OF MICROSECONDS
BETWEEN TRANSMISSION OF KEY EVENTS:
5000
```

Mit der Tastaturemulation hat sich noch einiges mehr getan. Wichtigster Punkt ist die geänderte Interpretation der Cursortasten. Diese liegen nun nicht mehr einfach "parallel" zum Zehnerblock, sondern arbeiten auch bei gedrücktem NumLock als Cursortasten. Die Erleichterungen beim Tippen sind dadurch immens. Die Helptaste ruft jetzt Hilfsfenster zum Programm auf.

Eine weitere Verbesserung fand bei den Copy- und Pastefunktionen statt. Die Zeichen, die sich beim IBM-Zeichensatz in der oberen Hälfte befinden, werden jetzt auch korrekt übertragen, falls es ein Pendant dazu im AMIGA-Zeichensatz gibt. Leider wird immer noch kein deutscher Tastaturtreiber unterstützt. Hat man diesen geladen, kommt es immer noch zu Fehlinterpretationen.

Eine für alle

Von vielen Benutzern wurde bisher eine Umsetzung der AMIGA-Maus auf den PC vermißt. Mit der Version 2.0 der Treibersoftware ist dies nun vorbei. Von nun ab können alle Programme für IBM-Kompatible, die eine Maus unterstützen, auch mit dieser bedient werden. Auf der AMIGA-Seite wird das Programm AMOUSE, das sich im PC-Ordner befindet, gestartet. Danach kann man unter MS-DOS einen der mitgelieferten Maustreiber laden. Hierunter findet man neben den normalen Maustreibern auch die für MS-Windows.

A PC
AMouse

Für den Betrieb mit der Sidecar hat man einen eigenen MS-Windows-Treiber entwickelt. Der normale Treiber *Amouse.com* kann mittels der Option "-x" an den Betrieb mit dem Veteranen angepaßt werden. Wer zwei Mäuse sein eigen nennt, kann sogar beide Mauszeiger gleichzeitig bewegen. Normalerweise wird dem PC nämlich der rechte Mausport zugewiesen. Eine Maus am linken Port wird vom AMIGA, eine zweite am rechten Eingang vom PC abgefragt (wobei hier der AMIGA natürlich auch seine Finger im Spiel hat). Damit man nicht zwingend zwei Mäuse benötigt oder ständig die Maus umstecken muß, hat man die Möglichkeit, die Zuordnung der Ports zu den Rechnern mittels AMIGA-P umzuschalten.

Im Betrieb gab es keine Probleme mit der Emulation. MS-Windows, GEM oder diverse Software, die mit der Maus bedient wird, alles funktionierte, ohne zu murren. Murren kann nur der Benutzer, weil die Maus soviel Platz braucht wie die AMIGA-Maus ohne Beschleuniger. Und wer benutzt die AMIGA-Maus noch ohne einen Speeder?

Der Sankt-Martins-PC

Wer was Warmes hat und auf jemanden trifft, der friert, soll mit dem armen Tropf teilen. Auch unser PC ist eine freundliche Natur und teilt seine hitzeabstrahlende, beruhigend säuselnde (es soll aber Menschen geben, denen dieses Geräusch mißfällt) Festplatte schon seit geraumer Zeit mit dem AMIGA. Bisher kamen aber die Daten recht langsam herüber, was nicht das erwartete Feuer beim Einlesen und Schreiben aufkommen ließ. Dem noch leicht fröstelnden AMIGA werden jetzt aber die Schweißperlen von der Stirn rinnen ob der schnellen Lieferung von Daten, die nun möglich ist. Die neue Software unterstützt nämlich das FastFileSystem für Janus-Partitionen. Ruft man das Kommando

```
DJMount FFS
```

auf, bindet der AMIGA die Festplatte als FFS-Device ins System ein. Will man vom alten Filingsystem auf das FFS umsteigen, ist aber eine Neuformatierung notwendig, was mit etwas Arbeit verbunden ist (Daten sichern, neu formatieren, Daten zurückspeichern). Zum Formatieren benötigt man jetzt nicht mehr einen extra Formatbefehl (DPFormat), der normale des Betriebssystem (V1.3) kann nun auch Janus-Platten handhaben. Beachten sollte man, daß beim Formatieren der Schalter "FFS" nicht benutzt werden darf. Format schreibt FFS-Janus-Devices automatisch im neuen Format, die Angabe von "FFS" führt zu einem Deadlock des Formatbefehls (bei den mir zur Verfügung stehenden Rechnern jedenfalls).

A PC
PCHard

Eine weitere Änderung am DJMount-Befehl ist wohl mehr kosmetischer Natur. Da der PC erst nach dem Befehl "BindDrivers" seinen Selbsttest ausführt und dann bootet, mußte bisher ein Wait-Befehl eingefügt werden, damit der AMIGA erst dann seine Partitionen mountet, wenn der PC betriebsbereit ist. Die Warteschleife ist nun in DJMount eingebaut, was den externen Waitbefehl überflüssig macht.

Die Treiber für die virtuellen PC-Laufwerke blieben funktionell unverändert. Da diese den Nachteil haben, daß der PC nicht von ihnen gebootet werden kann, hat man sich etwas Neues einfallen lassen, was nachfolgend erläutert werden soll.

Vortäuschung falscher Tatsachen - Der AMIGA geht straffrei aus

Hatte man bisher das unbedingte Verlangen, beide Rechner ohne Disketten hochzufahren, war man gezwungen, zwei Festplatten anzuschaffen. Dies ließ den Käufer in unermeßliche finanzielle Schwierigkeiten kommen, da hierzu Ausgaben in Höhe des Monatsbedarfs einer 5köpfigen Familie notwendig waren. Mit der neuen Treibersoftware ist das nun anders geworden. Die Software stellt einen Service zur Verfügung, der es ermöglicht, mit einer AMIGA-Festplatte eine bootfähige PC-Festplatte zu simulieren. Dies geht bis zur BIOS-Ebene herab, so daß sogar diverse Festplatten-Utilities wie z.B. Compress von PCTools arbeiten, die sich wirklich auf den tiefsten Ebenen mit der Harddisk unterhalten.

Die Emulation der Festplatte funktioniert in ähnlicher Weise wie bei JLink durch eine Datei auf einem AMIGA-Speichermedium. Dies muß noch nicht einmal unbedingt eine Festplatte sein. Versuche ergaben, daß es ebensogut mit der RAM-Disk oder einer normalen Diskette funktioniert. Daß deren Verwendung nicht sehr sinnvoll ist, versteht sich von selbst.

```
PC Display Screen
PC Color Display
Copyright (C) 1986 Commodore Electronics Limited

AMIGA PARTITION SETUP V1.2 :

Current drive :  00        Total space :   409  cylinders

Partition   Status    Type      Start     End     Size
    1         A       DOS         0       408     409



< 1 > : change current drive number.
< 2 > : change partition status.
< 3 > : create a new AMIGA partition.
< 4 > : delete an AMIGA partition.
<ESC> : return to DOS.
```

Bitte besuchen Sie uns in Halle 7 / Stand E 46
HANNOVER MESSE CeBIT '89
Welt-Centrum Büro · Information · Telekommunikation
8. – 15. MÄRZ 1989

DAS GFA-BASIC 3.0 BUCH

Lieferbar!

KLAUS SCHNEIDER,
OLIVER STEINMEIER,
PETER FRITZEN

Als optimale Ergänzung zum Handbuch des neuen GFA-BASIC 3.0-Interpreters bietet sich dieses Buch an. In zwei Hauptteilen wird zunächst eine systematische Einführung in die Programmierung von BASIC unter Berücksichtigung der besonderen Fähigkeiten von GFA-BASIC auf dem Amiga gegeben. Hier werden dem Neuling vom ersten Einzeiler bis zu abstrakten Datentypen alle Möglichkeiten der strukturierten Programmierung mit zahlreichen, durch Flußdiagramme transparenter gemachten Beispielen nahegebracht. Doch auch BASIC-erfahrene Programmierer lernen hier die neuen Strukturen kennen, die sich doch sehr von denen anderer Dialekte unterscheiden.

Der zweite Teil bauf auf dem ersten auf und vermittelt weitere Kenntnisse der Programmierung, anhand von Programmen, die wiederum ausführlich beschrieben und erklärt sind. Hier seien ein leistungsfähiges Grafikprogramm sowie zahlreiche Beispiele zur Betriebssystemprogrammierung genannt.

Die Benutzung der verschiedenen Libraries und Intuition-Funktionen wird detailliert erklärt, so daß die Verwendung dieser Möglichkeiten in eigenen Programmen keine Schwierigkeiten bereitet.

Durch zahlreiche Anhänge – neben vielen Tabellen finden Sie auch ein sehr ausführliches Stichwortverzeichnis – wird das Buch optimal ergänzt und kann problemlos zum Nachschlagen von Details benutzt werden.

AUS DEM INHALT:

Erklärung der Schleifen- und Programmstrukturen
- FOR-NEXT, WHILE-WEND, REPEAT-UNTIL, DO-LOOP
- Prozeduren, Funktionsunterprogramme und Verzweigungen
- Rekursion
- Beispielprogramme

Variablentypen und Arrays
- numerische und Zeichenkettenvariablen
- Arrays zur Aufnahme großer Datenmengen

Multitasking in GFA-BASIC
- Reagieren auf Ereignisse
- Zeitabhängige Prozeduraufrufe

Programmentwicklung
- Programmplanung und -entwurf
- strukturierte Programmierung
- TOP-DOWN-Prinzip
- Fehlersuche
- Debugging-Möglichkeiten

Dateiverwaltung
- sequentielle Dateien
- Random-Access-Dateien
- Funktionen und Befehle zur Diskettenverwaltung

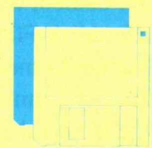
Grafikprogrammierung
- Die vielseitigen Möglichkeiten
- Turtlegrafik, Spriteprogrammierung
- Arbeiten mit mehreren Bildschirmen
- HAM-Modus
- Entwicklung eines vielseitigen Grafikprogramms

Sound- und Sprachprogrammierung
- Die Möglichkeiten der Befehle
- Verwendung der Anweisung in eigenen Programmen

Abstrakte Datentypen
- Verkettete Listen
- Bäume

Betriebssystem-programmierung
- Pulldownmenü-Steuerung
- Requester-Einsatz in Programmen
- Intuition und Graphics-Library
- Aufruf von Systemroutinen

Für Einsteiger ★ Fortgeschrittene ★ und Profis



ÜBER 500 Seiten EINSCHLIESSLICH PROGRAMMDISKETTE

Dem Buch liegt eine Programm-Diskette bei mit über 150 Übungs- und Beispielprogrammen

BUCH & DISKETTE KOMPLETT
HARDCOVER DM 59,-

Preise sind unverbindlich empfohlene Verkaufspreise

Heim Verlag
Heidelberger Landstr. 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

Schweiz: DataTrade AG
8021 Zürich
Langstr. 94

BESTELLCOUPON AMIGA GFA 3.0 BUCH

Bitte senden Sie mir ________ St. DAS GFA-BASIC **3.0 BUCH** einschließlich Programm-Diskette für DM **59,–** zzgl. DM 5,– Versandkosten (unabhängig von bestellter Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ________________________________

Straße, Hausnr. ________________ PLZ/Ort ________________

Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte.

Im Gegensatz zu den JLink-Drives ist zur Benutzung des Festplattenersatzes einiges an Vorarbeit notwendig. Durch die gute Benutzerführung der Dienstprogramme sind diese Vorbereitungen aber leicht durchzuführen.
Erster Schritt ist es, die Datei zu erzeugen, die die Festplatte emulieren soll. Dazu findet man das Programm MAKEAB im c-Ordner der Workbench. Startet man dieses Programm, wird man nach den Festplattenparametern wie Anzahl der Köpfe, Sektorenzahl pro Spur und Spuranzahl gefragt. Aus diesen Angaben ergibt sich unmittelbar die Speicherkapazität der Pseudofestplatte. Diese wird angezeigt, und nach einer Sicherheitsabfrage richtet der AMIGA die Datei ein. Da dieses File sofort in voller Größe eingerichtet wird (im Gegensatz zu den virtuellen Laufwerken von Jlink), muß natürlich genügend Platz auf dem Speichermedium frei sein. Bei der Plazierung des Files ist eine Besonderheit zu beachten. Wird es auf einer FFS-Partition angelegt, muß diese im Mountlisteintrag die Option Mask = 0 enthalten. Damit wird die Datenübertragung von Platte zum Speicher mittels DMA abgeschaltet, was die Geschwindigkeit der Platte herabsetzt. Commodore empfiehlt deshalb, für die Festplattendatei eine eigene Partition anzulegen, die dann alleine, ohne DMA-Transfer bedient wird.
Hat man das Harddisk-File eingerichtet, braucht man nur noch eine Datei namens *ABOOT.CTRL* im Ordner *SYS:PC\SYSTEM* anzulegen, die den Pfad, unter dem die Datei zu finden ist, enthält. Hat man zum Beispiel mit *Makeab* die Datei *PCPlatte* im Verzeichnis *DH1:IBM* erzeugt, muß Aboot.ctrl den Inhalt "*DH1:IBM/PCPlatte*" haben. Damit wären alle Dateien erzeugt, welche der AMIGA für diesen Trickbetrug benötigt.
In der Startup-Sequence muß sich an den Befehl

```
BindDrivers
```

direkt die Direktive

```
Run >NIL: PC/PCDisk
```

anschließen, da *PCDisk* vom AMIGA benötigt wird, um dem PC die Festplatte anzubieten. Ein Aufruf an späterer Stelle ist nicht möglich, da der PC dann schon in seiner Boot-Sequence zu weit fortgeschritten ist, um die Festplatte noch zu erkennen. Öffnet man nun ein PC-Fenster, findet man den PC in dem guten Glauben, er besäße eine Festplatte. Diese ist unter dem Buchstaben C erreichbar und kann nun partitioniert und mit *FORMAT C:/S* zur bootfähigen Festplatte gemacht werden. Von nun an kann man sie wie jede normale Festplatte benutzen. Die Geschwindigkeit liegt in etwa in den selben Bereichen. Dies hängt natürlich von der Geschwindigkeit der Platte selbst ab und liegt etwas unter der Performance, welche die Platte bei direktem Anschluß an den PC erreichen würde.

A PC
PCDisk

Aread und Awrite - aufgeblasen

Die Utilities *Aread* und *Awrite* waren schon in der alten Version der Janus-Software enthalten. Sie wurden aber kräftig überarbeitet und erweitert, was eine genauere Betrachtung für angebracht erscheinen läßt.
Aread und *Awrite* dienen zur Dateiübertragung zwischen beiden Rechnern. Voraussetzung für das Arbeiten der beiden Programme ist das Programm *PCDisk*, das auf dem AMIGA laufen muß. *Awrite* dient zur Dateiübertragung vom PC zum AMIGA. Die Syntax des Befehls ist:

```
AWRITE PCfilename Amiga-
filename [/B] [/NC] [/CR]
```

```
AWRITE PC-wild-card Amiga-
Directory [/B] [/NC] [/CR]
```

Es ist nun also möglich, mehrere Dateien mit einem Befehl zu übertragen. Die Optionen haben folgende Wirkung:

/B: Die Datei wird binär übertragen. Es erfolgen keine Konvertierungen von ASCII-Codes.

/NC: Es findet keine Übersetzung von ASCII-Codes des PC in die des AMIGA statt.

/CR: Schaltet die Übersetzung von CR+LF in LF ab.

"*/B*" ist äquivalent der Benutzung von "*/NC /CR*". Standardeinstellung ist die vollständige Konvertierung.

In die entgegengesetzte Richtung überträgt *Aread*. Die Syntax

```
AREAD Amigafilename PC-
   filename [/B] [/NC] [/CR]
```

ist der von *Awrite* ähnlich, unterstützt aber keine Wildcards. Die Optionen haben die gleiche Wirkung, die ASCII-Übersetzung benutzt aber eine andere Tabelle, um die AMIGA-Sonderzeichen auf den PC-Zeichensatz umzusetzen.

Dem Glücklichen schlägt eine Stunde

War bisher der PC im AMIGA in den unendlichen Tiefen des 1. Januar 1980 gefangen, so hat sich das geändert. Auch Commodore hat festgestellt, daß wir uns im Jahre 1989 befinden, und daß die wenigsten Benutzer ihr Gerät um Mitternacht einschalten. Die mitgelieferten Disketten enthalten nun 2 Programme, die dazu dienen, die Systemzeit vom AMIGA zum PC zu übertragen. Auf der AMIGA-Seite ist dies *TimeServ* in der Services-Schublade, das resident installiert wird, wenn man es aufruft. Auf dem PC braucht man dann nur *Atime* zu starten, und schon hat man auf beiden Rechnern die gleiche Uhrzeit. Ruft man *Atime* mit dem Schalter "/w" auf, wartet das Programm bei nicht laufendem *Timeserv*, bis dieses gestartet wird.

A PC
TimeServ

Das wäre also alles Neue, was die Janus-Software dem Anwender zu bieten hat. Die Veränderungen und Verbesserungen sind schon gewaltig. Die neue Software hat aber noch einen anderen Aspekt: Die Hinzunahme der Services. Hier bietet sich ein weites Feld für die Programmierer, zusätzliche Utilities, Emulationen und Anwendungen zu erstellen. Man darf also gespannt sein.

ENDE

KICK ME, AMIGA

R.C.T.

KICK

Requester und Menü Construction Tool für den AMIGA

Der Weg in völlig neue Dimensionen

Wollten Sie nicht schon immer ihre Programme mit einer professionellen Benutzerführung versehen?

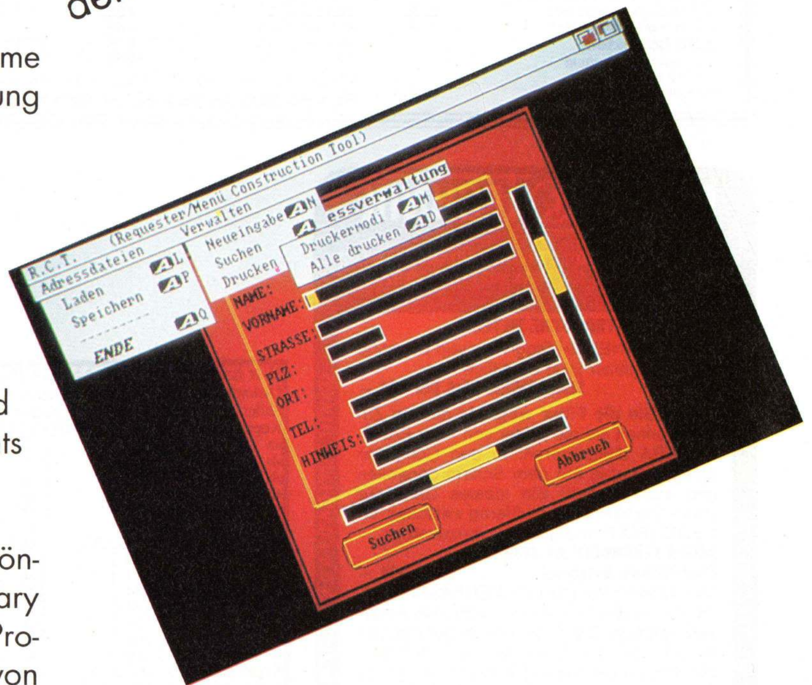
Mit dem R.C.T. ist das kein Problem, egal welche Programmiersprache Sie wählen.

Das Erstellen einer grafikorientierten Oberfläche ist für viele ein Buch mit sieben Siegeln. Das R.C.T. hilft Ihnen, Requester mit Gadgets und Images oder Menüs in verschiedenen Schriftfonts zu konstruieren.

Die konstruierten Requester und Menüleisten können mit den Funktionen der mitgelieferten Library verwaltet werden. Die Library kann von allen Programmiersprachen genutzt werden, auch von Basic aus.

Bei Programmiersprachen wie C oder Assembler besteht die Möglichkeit, die Funktionen der Library direkt an das Programm zu linken.

Der Sourcecode-Generator ermöglicht die Erzeugung von C-, Assembler- und GFA-BASIC-Source von den konstruierten Requestern und Menüleisten.



R.C.T. DM 129.-

HALLE 7 / E 42
HANNOVER MESSE CeBIT '89
Welt-Centrum Büro · Information · Telekommunikation
8. – 15. MÄRZ 1989

Bestellcoupon

Bitte senden Sie mir:

☐ R.C.T. DM 129.-

zuzüglich Versandkosten:
Inland DM 7.50
Ausland DM 10.-

Name

Straße

Ort

Datum Unterschrift

Auslandsbestellungen nur gegen Vorauskasse
Bei Nachnahme zuzüglich DM 3.50 Nachnahmegebühr

MAXON-Computer / Industriestr. 26 / 6236 Eschborn

VON ANDREAS BIERNAT

Die Floppy im Griff!!

Teil 1:
Die Einführung

Was, schon wieder ein Kurs Über das Trackdisk-Device? Viele von Ihnen werden nun wieder einen Artikel erwarten, der in kurzen Sätzen zeigt, wie man mit diesem Device von Diskette lesen und schreiben kann und der ein paar Daten aus verschiedenen 'Include-Files' in Tabellen darstellt. Dieser Kurs aber soll unter anderem Grundlagen und Zusammenhänge der Trackdisk-Devices klären, wie z.B. folgende Fragen: Was ist überhaupt ein Device? Was kann und muß man alles mit den Messages (Nachrichten) machen? Port, IO-Routinen - was ist das? Braucht man das? Neben der Theorie wird natürlich auch speziell auf das Trackdisk-Device eingegangen, d.h. Befehle werden erklärt mit Beispielen in Assembler.

Die Beispiele sind mit dem SEKA-Assembler geschrieben, können aber leicht auf andere Assembler, so z.B. den neuen KICK-ASS, angepaßt werden. Jemand der sich nicht mit 'Include-Files' auskennt, kann nämlich mit einer Anweisung wie *'INCLUDE "devices/trackdisk.i"'* nichts anfangen und wundert sich dann über Konstanten wie NUMSECS oder TD_SECSHIFT im Listing. Zu Beginn sollen einige theoretische Grundlagen behandelt werden , danach werde ich zur direkten Programmierung der Floppy kommen.

Device?

Ein Device ist ein Programmpaket, das auf niedrigster Ebene den Umgang mit der Hardware des AMIGA regelt. Der Ausdruck "auf niedrigster Ebene" läßt sich am besten an Abbildung 1 verdeutlichen.

Ein Device ist dabei die unterste Schnittstelle zur AMIGA-Hardware. Wie auch die Libraries verfügt jedes Device über eine Sprungtabelle mit negativen Offsets. Ein Device spricht man aber normalerweise nicht über eine Basisadresse

an wie eine Library, sondern man führt sogenannte IO-Funktionen (Input-Output) durch. Mittels dieser IO-Funktionen kommuniziert man mit dem Device. Die IO-Funktionen benötigen als Parameter immer eine sogenannte IO-Request-Struktur. Diese Struktur enthält alle nötigen Daten und Informationen, um mit dem Device arbeiten zu können. Sie hat folgendes Aussehen (siehe Abb. 2).

So, da wäre ja einiges zu erklären. Wir wollen alles der Reihe nach durchgehen. Der nun folgende Abschnitt Über das Message-System besteht aus viel Theorie. Es macht nichts, wenn man das eine oder andere nicht verstanden hat, weil das Verständnis dieses Abschnitts kein Muß für den späteren Umgang mit den Devices darstellt.

Zuerst wäre da die Message-Struktur. Es wurde ja bereits gesagt, daß mit Hilfe der IO-Request-Struktur kommuniziert wird. Kommunizieren heißt in diesem Fall Senden und Empfangen von Nachrichten (engl.: messages). Eine Message besteht zum einen aus einer Message-Struktur und der Nachricht selbst. Die Nachricht wird direkt an die Message-Struktur angehängt, was bedeutet, daß der Rest unserer IO-Request-Struktur (ab Offset 20) zu der Nachricht gehört. Die Message-Struktur sehen Sie in Abb. 3.

Die Node-Struktur am Anfang dient dazu, die Message in die Liste der empfangenen Messages des Message-Ports einzugliedern. Für die einfache Benutzung eines Devices hat man mit der Listenverwaltung nichts zu tun, aber der Vollständigkeit halber sei hier auch die Node-Struktur erklärt (siehe Abb. 4).

Die ersten beiden Zeiger dienen dem Durchsuchen der Liste, um von einem Eintrag (auch Node genannt) zum anderen zu kommen. Den Typ stellt eine Zahl dar, die aus Tabelle 1 abgelesen werden kann.

Message-Port

Der Message-Port ist eine Sammelstelle für Messages an einen Task. Bei der Handhabung mit Devices sind dabei die Messages von dem Device zu unserem eigenen Programm, also an unseren eigenen Task gemeint. Diese Sammel-

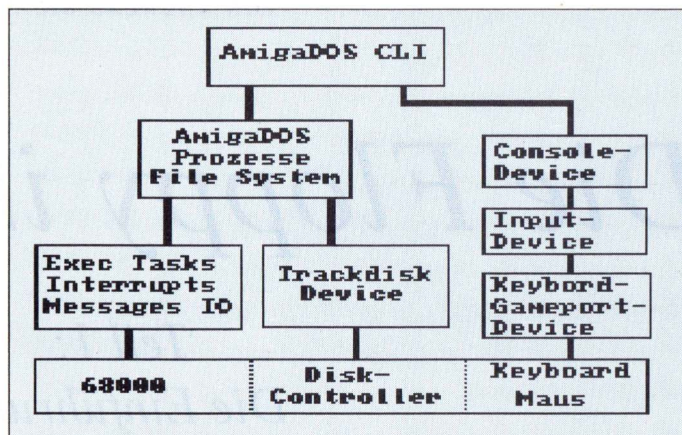


Abbildung1: Das Device - die Schnittstelle zur Hardware

Offset	Größe	Name	Beschreibung
0	--	Message	Message-Struktur
20	dc.l	Device	Zeiger auf Device-Struktur
24	dc.l	Unit	Zeiger auf Unit-Struktur
28	dc.w	Command	Wort, in dem der auszuführende Befehl steht
30	dc.b	Flags	Übergabe von device-spezifischen Statusmeldungen
31	dc.b	Error	Übergabe von Fehlermeldungen an den Programmierer

```
struct IORequest
{
struct Message io_Message;
struct Device *io_Device;
struct Unit *io_Unit;
UWORD io_Command;
UBYTE io_Flags;
BYTE io_Error;
};
```

Abbildung 2: Die IO-Request-Struktur

Offset	Größe	Name	Beschreibung
0	--	Node	Listeneintrag für eine Liste der empfangenen Nachrichten.
14	dc.l	ReplyPort	Zeiger auf Message-Port für Antwort-Nachricht
18	dc.w	Length	Länge der Message in Bytes

```
struct Message
{
struct Node mn_Node;
struct MsgPort *mn_ReplyPort;
UWORD mn_Length;
};
```

Abbildung 3: Die Message-Struktur

Offset	Größe	Name	Beschreibung
0	dc.l	Successor	Zeiger auf nachfolgenden Listeneintrag
4	dc.l	Predecessor	Zeiger auf vorherigen Listeneintrag
8	dc.b	Type	Type des Eintrages
9	dc.b	Pri	Priorität des Eintrages
10	dc.l	Name	Zeiger auf mit einem Nullbyte abgeschlossenen Namen des Eintrages

```
struct Node
{
struct Node *ln_Succ;
struct Node *ln_Pred;
UBYTE ln_Type;
BYTE ln_Pri;
char *ln_Name;
};
```

Abbildung 4: Die Node-Struktur

stelle wird wieder einmal durch eine Struktur dargestellt (siehe Abb. 5).

Die Node-Struktur am Anfang ist die einer Liste, mit deren Hilfe die Message-Ports verwaltet werden. Im *Name*-Feld ist ein Zeiger auf den Namen dieses Message-Ports gespeichert.
Die unteren beiden Bits im Flag-Byte bestimmen, was passiert, wenn der Message-Port eine Nachricht empfängt. Die Nachricht kann ignoriert oder es kann ein Software-Interrupt ausgelöst werden, oder man sendet ein Signal an den Ziel-Task.
Dabei wird ein vorher definiertes Bit in einer 32 Bit-Signalmaske in der Task-Struktur gesetzt. Das Signalbit ist durch eine Nummer zwischen 0 und 31 definiert. Diese Nummer steht in dem Byte "SigBit" (Offset 15). Bei der Arbeit mit Devices ist der Empfänger-Task unser eigener Task. Die Struktur des Listenkopfes der Messagelist ähnelt der Node-Struktur. In dieser Liste stehen alle empfangenen Nachrichten.

Soviel nun zu dem Message-System. Wie schon eingangs gesagt, muß man diesen Abschnitt nicht unbedingt verstanden haben, um den Kurs fortzuführen. Wir kehren an dieser Stelle also zu unserer IO-Request-Struktur zurück. Als ersten Teil der Message finden wir dort einen Zeiger auf eine Device-Struktur. Es wurde am Anfang gesagt, daß ein Device über eine Sprungtabelle mit negativen Offsets verfügt (wie eine Library). Die so erreichbaren Funktionen dienen zum Öffnen und Schließen eines Devices sowie zum Ausführen einer IO-Operation. Diese Funktionen sind nötig, damit eine Funktion wie 'OpenDevice()' jedes Device öffnen kann, auch wenn die dafür zu erledigenden Tätigkeiten von Device zu Device verschieden sind. Die *OpenDevice()*-Funktion aus der Exec-Library springt also in die Open-Funktion des gewünschten Devices ein, um so alle notwendigen Aufgaben zu erfüllen. Die wichtigsten Funktionen, die jedes Device zur Verfügung stellt, sind:

Offset	*Funktion*
-36	AbortIO
-30	BeginIO
-12	Close
-6	Open

Task	1	
Interrupt	2	Die Priorität kann zwischen -128 u. +127 schwanken,
Device	3	wobei +127 die höchste und -128 die niedrigste
MsgPort	4	Priorität bedeutet. Einige Listen werden nach der
Message	5	Höhe der Priorität sortiert.
FreeMsg	6	
ReplyMsg	7	Der Name soll die Node (den Listeneintrag) kenn-
Resource	8	zeichnen, und zwar am besten so, daß man schon am
Library	9	Namen erkennen kann, um was für eine Node es sich
Memory	10	handelt.
Softint	11	
Font	12	Mit Hilfe dieser Node kann man zu anderen Nodes
Process	13	gelangen (über die beiden Zeiger am Anfang) und so
Semaphore	14	feststellen, welche weiteren Messages dieser Port empfangen hat.

Tabelle 1

Offset	*Größe*	*Name*	*Beschreibung*
0	--	Node	Listeneintrag fÜr Liste der Message-Ports
14	dc.b	Flags	Flags für Aktionsmodus
15	dc.b	SigBit	Signalbit des Tasks
16	dc.l	SigTask	Zeiger auf Empfänger-Task
20	--	MsgList	Listenkopf der Messageliste

```
struct MsgPort
{
struct Node mp_Node;
UBYTE mp_Flags;
UBYTE mp_SigBit;
struct Task *mp_SigTask;
struct List mp_MsgList;
};
```

Abbildung 5: Die Message-Port-Struktur

Befehl	*Command-Zahl*	*Beschreibung*
Invalid	0	ungültiges Kommando
Reset	1	Zurücksetzen des Devices in den Anfangszustand
Read	2	Lesen aus dem Device
Write	3	Schreiben in das Device
Update	4	Aufarbeiten der Puffer
Clear	5	Löschen aller Puffer
Stop	6	Pause einlegen
Start	7	Nach der Pause weitermachen
Flush	8	Abbruch der momentanen Arbeit

Abbildung 6: Die Device-Struktur

Wenn ein Device geöffnet werden soll, wird der Zeiger vom System auf die Device-Struktur gesetzt. Sie wird auch hauptsächlich vom System benutzt. Wenn wir ein Device öffnen wollen, haben wir mit diesem Eintrag der IO-Request-Struktur noch nichts zu tun. Ähnliches gilt für den Zeiger auf die Unit-Struktur. Dieser Zeiger ist ein privates Element und deviceabhängig. Er wird beim Öffnen des Devices vom System gesetzt und sollte vom Programmierer nicht benutzt werden. Der nächste Eintrag in der IO-Request-Struktur ist ein Wort, in dem ein Befehl des Devices steht, der ausgeführt werden soll. Der Befehl wird von einer Zahl repräsentiert, die aus einer Tabelle zu entnehmen ist. Jedes Device verfügt über eine Anzahl von Standardbefehlen. Diese sind obiger Abb. 6 zu entnehmen.

Von den Standardkommandos für jedes Device wollen wir nun übergehen zu den gültigen Trackdiskkommandos (siehe Abb. 7).

Dazu kommt noch eine Liste von erweiterten Kommandos, die die gleiche Funktion wie obige Kommandos haben, jedoch nur, wenn kein Diskettenwechsel vorgenommen wurde

Befehl	*Command-Zahl*
Read	32770 ($8002)
Write	32771 ($8003)
Update	32772 ($8004)
Clear	32773 ($8005)
Motor	32777 ($8009)
Seek	32778 ($800a)
Format	32779 ($800b)
Rawread	32784 ($8010)
Rawwrite	32785 ($8011)

Wenn man also das Kommando *Lesen* ausführen lassen will, schreibt man mit einem einfachen MOVE-Befehl die Zahl 2 in das Kommandofeld der IO-Request-Struktur. Zeigt z.B. A1 auf unsere IO-Request-Struktur, dann ist mit einem

```
MOVE #2,28(A1)
```

alles getan. Das nächste Element der IO-Request-Struktur ist ein Flagbyte. Es wird benötigt, um devicespezifische Statusmeldungen oder Befehle übergeben zu können. Das Byte ist in High- und Low-Nibble unterteilt. Die unteren vier Bits werden von Exec für interne Zwecke benutzt, während die oberen vier vom Programmierer benutzt werden können, um mit dem Device zu kommunizieren. Von dieser Möglichkeit wird aber wenig Gebrauch gemacht. Das Error-Byte der IO-Request-Struktur übergibt Fehlermeldungen an den Programmierer. Die möglichen Werte für das Trackdisk-Device entnehmen sie bitte Abb. 8.

Soweit die Erklärung der IO-Request-Struktur. Das war aber leider noch nicht alles. Denn oft reicht diese Struktur nicht aus, um ein Device benutzen zu können. Für solch einen Fall existiert noch eine weitere Struktur, die dem Benutzer mehr Möglichkeiten bietet. Sie nennt sich IO-Standardrequest-Struktur (Abk: IOStdReq). Sie besteht aus der IO-Request-Struktur und vier weiteren Einträgen (siehe Abb. 9).

Das Langwort *Actual* enthält die Anzahl Bytes, die ordnungsgemäß übertragen wurden. Es kann erst nach Beendigung der Übertragung ausgelesen werden (z.B. mit *MOVE.L 32(A1),D0*).

In dem Feld *Length* wird die Länge der zu übertragenden Daten in Bytes angegeben. Da die Floppy nur ganze Sektoren lesen kann, muß die Länge ein ganzzahliges Vielfaches von 512 sein, da ein Sektor 512 Bytes lang ist. *Data* ist ein Zeiger auf den Speicherbereich, in den die zu übertragenden Daten hineingeschrieben werden sollen. Durch den Offset wird die Position der Quelldaten bestimmt. Da ein Datenbereich auf Diskette nur an einem bestimmten Sektor (oder Block) anfangen kann, muß auch der Offset ein ganzzahliges Vielfaches von 512 sein. Man berechnet ihn man nach der Formel *Offset = Blocknummer*512* . Das Directory der Diskette hat beispielsweise die Blocknummer 880. Wenn man mit den erweiterten Kommandos der Trackdisk-Devices arbeiten will, muß die IOStdReq-Struktur noch um zwei Einträge verlängert werden. Vollständig ist sie in Abb. 10 zu sehen.

Das ist also unsere erweiterte IO-Request-Struktur, kurz *IOExtTD* (da diese Form der Struktur nur für das Trackdisk-Device gebraucht wird). Der Eintrag *Count* wird benötigt, um festzustellen, daß auch wirklich keine Diskette gewechselt wurde, was ja Bedingung für die erweiterten Kommandos ist. Der Programmierer benützt diesen Zähler aber selten.

Befehl	*Command*	*Beschreibung*
Read	2	Lesen eines oder mehrerer Sektoren von Diskette
Write	3	Schreiben eines oder mehrerer Sektoren auf Diskette
Update	4	Trackpuffer auf Diskette zurückschreiben
Clear	5	Trackpuffer für ungültig erklären
Motor	9	Ein-/Ausschalten des Laufwerksmotors
Seek	10	Schreib-/Lesekopf auf einen best. Track positionieren
Format	11	Initialisierung eines oder mehrerer Tracks
Remove	12	Interrupt-Routine installieren, die bei einem Diskettenwechsel aufgerufen wird
Changenum	13	Anzahl der Diskettenwechsel ermitteln
Changestate	14	Feststellen, ob eine Diskette eingelegt ist
Protstatus	15	Feststellen, ob eine Diskette schreibgeschützt ist
Rawread	16	Lesen des MFM-codierten Disketteninhalts
Rawwrite	17	Schreiben von MFM-codierten Daten auf Diskette
Getdrivetype	18	Laufwerkstyp ermitteln (1=3.5" / 2=5.25")
Getnumtracks	19	Gesamtzahl der Tracks ermitteln
Addchangeint	20	Interrupt-Routine installieren, die bei einem Diskettenwechsel aufgerufen wird (geht nicht!)
RemChangeint	21	Routine abschalten (nutzlos!)

Abbildung 7: Trackdiskkommandos

20	NotSpecified	unbekannter Fehler
21	NoSecHdr	kein Sektor-Header vorhanden
22	BadSecPreamble	ungültiger Sektorvorspann
23	BadSecID	ungültige Sektor-ID
24	BadHdrSum	falsche Header-Checksumme
25	BadSecSum	falsche Sektor-Checksumme
26	TooFewSecs	nicht genug Sektoren verfügbar
27	BadSecHdr	ungültiger Sektor-Header
28	WriteProt	Diskette schreibgeschützt
29	DiskChanged	Diskette ist gewechselt worden
30	SeekError	Track nicht gefunden
31	NoMem	nicht genug Speicher
32	BadUnitNum	ungültige Laufwerksnummer
33	BadDriveType	ungültiger Laufwerkstyp
34	DriveInUse	Laufwerk bereits aktiv
35	PostReset	Resetphase

Abbildung 8: Die möglichen Fehlermeldungen des Trackdisk-Devices

Offset	*Größe*	Name	Beschreibung
0	--	IOReq	IO-Request-Struktur (wie oben)
32	dc.l	Actual	Anzahl der fehlerlos übertragenen Bytes
36	dc.l	Length	Anzahl der zu übertragenden Bytes
40	dc.l	Data	Zeiger auf Puffer für zu übertragende Daten
44	dc.l	Offset	Offsetwert zum Datenanfang

```
struct IOStdReq
{
struct Message io_Message;
struct Device *io_Device;
struct Unit *io_Unit;
UWORD io_Command;
UBYTE io_Flags;
BYTE io_Error;
ULONG io_Actual;
ULONG io_Length;
APTR io_Data;
ULONG io_Offset;
};
```

Abbildung 9: IO-Standardrequest-Struktur

Der letzte Eintrag ist ein Zeiger auf einen Datenblock, der zusätzlich zu den eigentlichen Sektordaten die Sektor-Label-Daten enthält. Dies sind Kennungen zwischen den Sektoren in einem Track. Da das Trackdisk-Device immer trackweise mit der Floppy arbeitet, hat es diese Kennungen sowieso im Speicher.
Normalerweise sind diese Kennungen nicht nötig. Doch wozu brauchten wir die IO-Request-Struktur nochmal? Wir wollten mit ihrer Hilfe mit dem Device kommmunizieren. Das bedeutet Senden und Empfangen von Nachrichten. Diese Nachrichten können nun mit folgenden, verschiedenen IO-Funktionen (Input-Output) gehandhabt werden:

Teil der Exec-Library:

1) *DoIO*

Syntax: Fehler = DoIO(IORequest)
D0 -456 A1

Diese Funktion wird meistens für die Ein-/Ausgabesteuerung verwendet. Sie wartet, bis der übergebene Befehl beendet wird, und kehrt erst dann zum eigentlichen Programm zurück. Wenn D0 ungleich Null ist, ist ein Fehler aufgetreten.

Offset	*Größe*	*Name*	*Beschreibung*
0	--	IOStdReq	siehe oben
48	dc.l	Count	Zähler für Diskettenwechsel
52	dc.l	SecLabel	Zeiger auf Labelfeld (16 Bytes)

```
struct IOExtTD
{
struct IOStdReq iotd_Req;
ULONG iotd_Count;
ULONG iotd_SecLabel;
};
```

Abbildung 10: Die IO-Standardrequest-Struktur

2) *SendIO*

Syntax: SendIO(IORequest)
-462 A1

Die Funktion dient zum Senden eines IOs an das entsprechende Device, wartet jedoch nicht auf dessen Beendigung.

3) *CheckIO*

Syntax: fertig = CheckIO(IORequest)
D0 -468 A1

Die Funktion prüft, ob ein bestimmter IO-Prozeß bereits abgearbeitet wurde. Sollte dies der Fall sein, wird in D0 der Zeiger auf die entsprechende IO-Request-Struktur zurückgegeben, andernfalls eine Null.

4) *WaitIO*

Syntax: WaitIO(IORequest)
-474 A1

Die Funktion wartet so lange, bis der IO-Prozeß erledigt ist. Die Funktionen SendIO() und WaitIO() ergeben zusammengesetzt die Funktion DoIO().

5) *AbortIO*

Syntax: AbortIO(IORequest)
-480 A1

Die Funktion beendet einen IO-Prozeß.

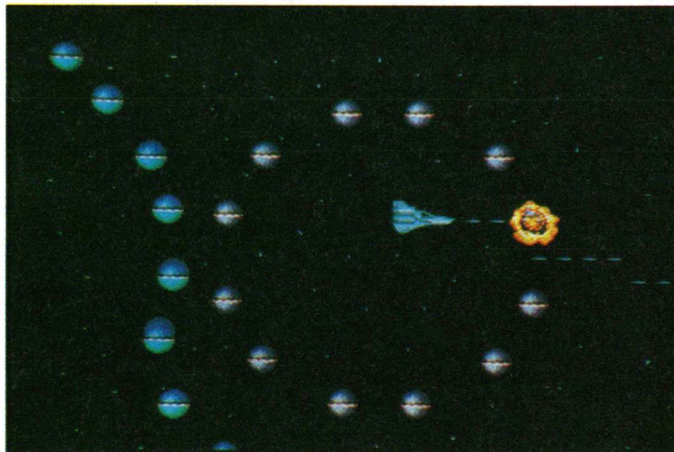
Das wär's für diesen Teil des Kurses - viel Theorie! Aber in der nächsten Ausgabe fängt die Programmierung ja schon an.

ENDE

MARTIN SILBERNAGL

KICKUP

Man faßt es nicht: endlich fertig oder hoffentlich



2765ster und letzter Teil

Natürlich hetz' ich dem Redaktionsschluß wieder seit 6 Tagen hinterher, was mich aber ebensowenig vom Artikelschreiben abhält, wie die Tatsache, daß meine Schwester mir meine letzten Toffifees geklaut hat.

Wie Sie als intelligenter Mitteleuropäer schon aus der Überschrift schließen konnten, geht unsere Serie mit diesem bombastischen Abschnitt zu Ende (oooooh!). Ich hatte in mein Leben nämlich noch andere Dinge eingeplant als mich immer mit dem KickUp 'rumzuärgern. Beziehungweise, was viel schlimmer ist, mit meinem Textprogramm: der Betarelease von Textcraft Plus, dem einzigen, das nicht abstürzt.

Aber bevor ich sentimental werde, bespreche ich ein letztes Mal das Programmstück, das neu dazugekommen ist. Voraussetzung ist natürlich, daß Sie das letzte Mal (und natürlich die Folge davor und die davor) alles (oder fast alles) verstanden haben.
Vorher sollte ich aber vielleicht noch sagen, daß man das Spiel zwar spielen kann, aber zur Vollendung fehlen noch der Sound (tippen Sie mal 'ne Explosion ab - meine kürzeste hatte 12 Kilobyte) und die Score-Anzeige (keine Lust mehr gehabt...) und vielleicht noch die Explosion des Raumschiffs - aber Sie wollten ja keine Daten-Sahara, sondern ein Programm, an dem man sieht, wie man (also ich natürlich) programmiert.

Mit dabei ist diesmal aber die Routine(n) zum Darstellen, Löschen und Bewegen der Feinde. Zuerst hatte ich hierfür ein Programmstück vorgesehen, das verschieden große Angreifer darstellen kann, das aber leider zu lang war. Deshalb hier nur eine kürzere Version für Angreifer und Explosionen gleicher Größe. Ob gleiche Größe oder nicht - natürlich interessiert Sie das Prinzip, nach dem die Angreifer verwaltet werden. Also gaaanz einfach. Nämlich fast genauso, wie die Schüsse. Nur gibt es da einen kleinen Unterschied: Die Liste mit den Schüssen ist immer durchgehend belegt, nur ihre Länge verändert sich. Um dieses Prinzip zu verwirklichen, war eine ziemlich komplizierte Routine nötig (ab MSO_COPY_LOOP). Und weil ich keinen Zopf hatte, dasselbe bei den Angreifern wiederholen zu müssen, dachte ich nach, wie man eine Liste noch verwalten könnte.
Ich kam darauf, daß ich ja auch eine zweite Liste führen könnte, in der vermerkt ist, ob ein Angreifer angezeigt wird oder nicht. Aber just in dem Moment, in dem ich die Sache verwirklichen wollte, fiel es mir wie Schuppen von den Augen: im Programm mußte doch sowieso eine Liste darüber geführt werden, welche der vier verschiedenen

Phasen der Angreifer gerade dargestellt wird. Diese könnte ich für diesen Zweck nutzen. Außerdem könnte sie noch angeben, in welcher Explosionsphase sich der Angreifer gerade befindet.

Also alles kein Problem: Im Speicher sind jetzt mehrere Listen. Die X-Positionen der Angreifer (EN_X), die Y-Positionen der Angreifer (EN_Y), die Adressen vor einem und vor zwei Videobildern (EN_LAST, EN_BE_LAST) und die eben angesprochene Animationsliste (EN_ANIM). Um dem Assembler mitzuteilen, wieviel Speicher die Listen brauchen (wie groß die Listen sein sollen), nehme ich die DS-Anweisung (alter Hut, oder?).

Easy run? Gar kein Zweifel

Schauen wir doch mal die Routine an, die die Raumschiffe bewegt: MOVE_ENEMYS. Zuerst ein JSR in ein Unterprogramm, das die Animationsphasen hochzählt und die Angreifer bewegt. Für den Inhalt der Listen ist also gesorgt. Angewandt werden sie, indem der Blitter nun entsprechend den Listen verwendet wird. Konkret bedeutet das, daß die X- und Y-Werte mit der Routine CALC_POS in Adressoffsets umgerechnet werden. Mit ihnen und dem Inhalt der FIELD_BASE1 wird die Adresse berechnet, an die geblittet wird. Mittels Schleife und WAIT_BLIT werden nun die fünf angezeigten Planes geblittet. Doch nach dieser Schleife und der Abfrage, ob der Angreifer gerade explodiert, kommt etwas ganz Neues (was aber trotzdem ziemlich wichtig ist): die Kollisionsabfrage.

Um pixelgenau Treffer von Nichttreffern unterscheiden zu können, brauchen wir natürlich wieder den Blitter (man sieht glaub' ich, warum das Blitter-Nasty-Bit gesetzt ist). Um aber die Abfrage wirklich zu verstehen, müssen Sie sich bitte daran erinnern, daß wir nicht nur fünf, sondern sieben Planes reserviert haben, wobei wir (oder halt der AMIGA) in die sechste die Schattenmasken des Raumschiffs und in die siebte die der Schüsse geblittet haben. Wir brauchen also nur die Schattenmaske eines Angreifers mit der Adresse zu verknüpfen (genauer gesagt: zu ANDen), an der er steht und zu prüfen, ob Einsen vorgekommen sind. Um uns eine

```
;  ------------------------------------------------------------
;
;            the official Kickstart-game, part five!
;
;            code and comments by Martin Silbernagl
;
;            graphics by Sebastian Dosch
;
;            last revision: 16/01/1989
;
;  ------------------------------------------------------------

[...]

TITLE_P   EQU      40*200            ; Größe einer Titelbild-Plane
TITLE     EQU      TITLE_P*5         ; Größe des Titelbildes
FIELD_B   EQU      48                ; Bytes pro Zeile beim Titelbild
FIELD_H   EQU      288+32            ; Höhe des Spielfeldes
FIELD_P   EQU      FIELD_B*FIELD_H   ; Größe einer Spielfeld-Plane
FIELD_O   EQU      FIELD_P*7         ; Größe eines Spielfeldes

[...]

YSPEED    EQU      3                 ; Geschwindigkeit des Raumschiffs
LENGTH    EQU      8192              ; Buffergröße festlegen
SHOOT_M   EQU      8                 ; Maximale Anzahl der Schüsse
SHOOT_W   EQU      2                 ; Breite des Schusses in Words

[...]

SHOOT_X   EQU      (SHIP_W-1)*16     ; rel. Pos. des Schusses: X-Wert
SHOOT_Y   EQU      SHIP_H/2          ; rel. Pos. des Schusses: Y-Wert
SHOOT_S   EQU      8                 ; Geschwindigkeit der Schüsse
SO_BLIT   EQU      SHOOT_H*64+SHOOT_W   ; Blitter"SIZE" des Schusses
ENEMY_N   EQU      4+7                  ; Anzahl der Angreiferphasen
ENEMY_W   EQU      2                 ; Breite des Angreifers in Words
ENEMY_B   EQU      ENEMY_W*2         ; Breite des Angreifers in Bytes
ENEMY_H   EQU      16                   ; Höhe des Angreifers
ENEMY_P   EQU      ENEMY_H*ENEMY_B      ; Größe einer Plane
EN_BLIT   EQU      ENEMY_H*64+ENEMY_W ; Blitter"SIZE" des Angreifers
EN_M      EQU      16                ; Maximale Anzahl der Angreifer
EXP_ST    EQU      6                 ; erste Explosionsanimationsphase
EXP_ED    EQU      EXP_ST+6          ; letzte Explosionsanimationsphase
MAXX_SO   EQU      8*(FIELD_B-SHOOT_B+2)-1 ; Maximaler X-Bereich des
                                             Schusses
MAXX      EQU      320               ; Maximale X-Position des Schiffs
MAXY      EQU      288               ; Maximale Y-Position des Schiffs
MINX      EQU      32                ; Minimale X-Position des Schiffs
MINY      EQU      32                ; Minimale Y-Position des Schiffs
BOBS      EQU      6*(SHIP_P+SHOOT_P+ENEMY_N*ENEMY_P)


START:
 MOVEM.L  D0-D7/A0-A6,-(A7)         ; Alle Register auf den Stack legen

[...]

 MOVE.B   (A0)+,(A1)+               ; ein Byte kopieren
 DBRA     D0,COPY_BOBS              ; Schleife machen

 MOVE.W   #ENEMY_N,D0               ; Anzahl der Eintragungen holen
 MOVE.L   BOB_BASE,D1               ; Adresse des Bob-Chip-Mems
 LEA      EN_GFX+4,A0               ; Adresse der Animationsadressen
ADD_BASE:
 ADD.L    D1,(A0)+                  ; Offset addieren
 DBRA     D0,ADD_BASE               ; Schleife machen

 MOVE.L   #ADSR1,D0                 ; Ende der letzten Wellenform
 SUB.L    #WAVE1,D0                 ; davon Anfang der ersten abziehen

[...]

 MOVE.W   #$C028,INTENA             ; PORTS- und VERB-Int. erlauben
 MOVE.W   #$7ff0,DMACON             ; Alle DMA ausschalten
 MOVE.W   #$830f,DMACON             ; DMAEN und BPLEN anschalten

 LEA      TITLE_CMAP,A0             ; Adresse der Quellcolormap
 JSR      COPY_COLORS               ; Farben setzen

[...]

 MOVE.W   #$7fff,INTENA             ; Alle Ints. ausschalten
 MOVE.W   #$C028,INTENA             ; PORTS- und VERB-Int. erlauben
 MOVE.W   #$7fff,DMACON             ; Alle DMA ausschalten
```

solche Abfrage zu erleichtern, gibt es im DMACONR-Register ein Bit, das angibt, ob bei einer Blitteroperation nur Nullen entstanden sind. Folglich prüfen wir das Bit und sehen, ob der Angreifer getroffen ist, oder ob der Angreifer das Raumschiff getroffen hat.

Gerade eben sprachen wir von einer Routine, die die Angreifer bewegt und animiert - das hat sich ja ganz schön angehört. Aber wie soll das gehen? Wie immer eigentlich ganz einfach, aber dieser Silbernagl muß ja immer alles so kompliziert machen:

Das Unterprogramm (INIT_ENEMYS1), in das gesprungen wird, wird nur einmal durchlaufen, denn es verbiegt den JSR auf die Routine MOVE_ENEMYS1, da es nur einmal gebraucht wird. Dies geschieht, weil es alle Register, mit denen die Routine MOVE_ENEMYS1 arbeitet, auf die Anfangswerte setzt. Da die Inhalte dieser Register aber weg wären, wenn das Programm weiterläuft, verwenden wir hier den MOVEM-Befehl (den kennen Sie vom Stack) etwas zweckentfremdet: Statt dem Stackpointer A7/A7' wird A6 verwendet, um einen Teil der Register immer wieder neu zu laden und einen anderen Teil als globale Variablen zu benutzen. Genial, was?

Ansonsten dürften sowohl in MOVE_ENEMYS1 als auch in MOVE_ENEMYS2 keine Besonder- oder Unklarheiten sein. Ach ja: Woher bekommt mam so eine Koordinatenliste, wie sie als zweite Formation vorkommt? Da gibt's nur wenige Möglichkeiten: Entweder Sie schreiben sich auch dafür ein Programm, oder Sie nehmen sich den Solartaschenrechner zur Hand.

So. Aus.

Es war schön mit Ihnen, vielleicht sehen wir uns auf der CES.

Ihr Martin.

```
 MOVE.W #$8740,DMACON          ; BLTPRI,DMAEN,BPLEN & BLTEN setzen
 LEA    MAIN_CMAP,A0           ; Adresse der Quellcolormap
[...]
 MOVE.W #$0000,BPLCON1         ; Scrollregister auf Null setzen
 MOVE.W #$0000,BPLCON2         ; Prioritäten normal setzen
 MOVE.W #4,BPL1MOD             ; Modulo setzen
 MOVE.W #4,BPL2MOD             ; Modulo setzen
 MOVE.W #$1671,DIWSTRT         ; linke, obere Ecke setzen
 MOVE.W #$36d1,DIWSTOP         ; rechte, untere Ecke setzen
 MOVE.W #$0030,DDFSTRT         ; data-fetch-start definieren
 MOVE.W #$00d8,DDFSTOP         ; data-fetch-start definieren
 MOVE.L #INIT_ENEMYS1,MOVE_ENEMYS+2 ; Vektor verbiegen
MAIN_LOOP:
WT_V:     JMP     W_OFF
W_ON:     MOVEQ   #2,D1
TST2:     MOVE.W  #$ffff,D0
TST1:     DBRA    D0,TST1
 DBRA     D1,TST2
W_OFF:
 MOVEA.L FIELD_BASE1,A0           ; Adresse des ersten Spielfeldes
 MOVE.L  FIELD_BASE2,FIELD_BASE1  ; Adresse des zweiten kopieren
 MOVE.L  A0,FIELD_BASE2           ; Vertauschung vollenden
 ;        MOVE.W #$f,COLOR00
 JSR      DEL_STARS               ; Sterne löschen
 ;        MOVE.W #$f0,COLOR00
 JSR      DEL_SHOOTS              ; Schüsse löschen
 ;        MOVE.W #$ff,COLOR00
 JSR      DEL_SHIP                ; Schiff löschen
 ;        MOVE.W #$f00,COLOR00
 JSR      DEL_ENEMYS              ; Angreifer löschen
 ;        MOVE.W #$f0f,COLOR00
 JSR      MOVE_STARS              ; Sterne bewegen
 ;        MOVE.W #$ff0,COLOR00
 JSR      MOVE_SHOOTS             ; Schüsse bewegen
 ;        MOVE.W #$fff,COLOR00
 JSR      MOVE_SHIP               ; Schiff bewegen
 ;        MOVE.W #$08f,COLOR00
 JSR      MOVE_ENEMYS             ; Angreifer bewegen
 ;        MOVE.W #$0,COLOR00
 MOVE.L  #MAIN_WAIT,MAIN_WAIT+2 ; Vektor setzen
MAIN_WAIT:
 JMP      MAIN_WAIT               ; auf VERTB warten
MAIN_JMP:
 JMP      MAIN_LOOP               ; Hauptschleife
[...]
ES:                               ; Wird bei esc (=$45) aufgerufen
 JMP      ENDING                  ; beenden
SD:                               ; Wird bei Shift down aufgerufen
 MOVE.L   #W_ON,WT_V+2            ; Auf Langsam stellen
 ADDQ.B   #1,SHIFT                ; Shift-Zaehler um eins erhoehen
 JMP      END_PORTS               ; und zurueck
SU:                               ; Wird bei Shift release aufgerufen
 MOVE.L   #W_OFF,WT_V+2           ; Auf Schnell stellen
 SUBQ.B   #1,SHIFT                ; Shift-Zaehler um eins erniedrigen
 JMP      END_PORTS               ; und zurueck
[...]
 MOVEM.L (A7)+,D0/A0-A1           ; Register retten
 RTS
WAIT_JOYSTICK:                    ; auf Knopfdruck warten
 BTST     #6,CIAA_PRA             ; Mausknopf gedrückt?
 BEQ.S    PRESSED                 ; wenn nicht, wieder prüfen
 BTST     #7,CIAA_PRA             ; Joystickknopf gedrückt?
 BNE.S    WAIT_JOYSTICK           ; wenn nicht, wieder prüfen
PRESSED:
 RTS
```

```

[...]

TST_X:
 MOVE.L  SH_X,D0          ; X-Koordinate nach D0
 CMP.L   #MINX,D0         ; kleiner als erlaubt?
 BGE.S   NOT_NEGX         ; wenn nicht, dann springen
 MOVE.L  #MINX,SH_X       ; auf kleinsten Wert setzen

[...]

TST_Y:
 MOVE.L  SH_Y,D0          ; Y-Koordinate nach D0
 CMP.L   #MINY,D0         ; kleiner als erlaubt?
 BGE.S   NOT_NEGY         ; wenn nicht, dann springen
 MOVE.L  #MINY,SH_Y       ; auf kleinsten Wert setzen

[...]

SHOOT_DATA:
 dc.w    $d3d3,$0000
 dc.w    $e1e1,$0000
 dc.w    $fbfb,$0000
 dc.w    $fbfb,$0000
 dc.w    $0404,$0000
 dc.w    $ffff,$0000

ENEMY_A1:
 dc.w    $0000,$0000
 dc.w    $03e0,$0000
 dc.w    $0c38,$0000
 dc.w    $130c,$0000
 dc.w    $2dc6,$0000
 dc.w    $18c6,$0000
 dc.w    $50e3,$0000
 dc.w    $6f1f,$0000
 dc.w    $5555,$0000
 dc.w    $2019,$0000
 dc.w    $50e7,$0000
 dc.w    $18c6,$0000
 dc.w    $2dce,$0000
 dc.w    $131c,$0000
 dc.w    $0c78,$0000
 dc.w    $03e0,$0000

 dc.w    $0000,$0000
 dc.w    $03e0,$0000
 dc.w    $0ff8,$0000
 dc.w    $1cfc,$0000
 dc.w    $303e,$0000
 dc.w    $203e,$0000
 dc.w    $601f,$0000
 dc.w    $70ff,$0000
 dc.w    $2222,$0000
 dc.w    $4007,$0000
 dc.w    $601f,$0000
 dc.w    $203e,$0000
 dc.w    $303e,$0000
 dc.w    $1cfc,$0000
 dc.w    $0ff8,$0000
 dc.w    $03e0,$0000

 dc.w    $0000,$0000
 dc.w    $03e0,$0000
 dc.w    $0ff8,$0000
 dc.w    $1ffc,$0000
 dc.w    $3ffe,$0000
 dc.w    $3ffe,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000

 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $3ffe,$0000
 dc.w    $3ffe,$0000
 dc.w    $1ffc,$0000
 dc.w    $0ff8,$0000
 dc.w    $03e0,$0000

 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $7fff,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000

 dc.w    $0000,$0000
 dc.w    $03e0,$0000
 dc.w    $0ff8,$0000
 dc.w    $1ffc,$0000
 dc.w    $3ffe,$0000
 dc.w    $3ffe,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $3ffe,$0000
 dc.w    $3ffe,$0000
 dc.w    $1ffc,$0000
 dc.w    $0ff8,$0000
 dc.w    $03e0,$0000

ENEMY_A2:
 dc.w    $0000,$0000
 dc.w    $03e0,$0000
 dc.w    $0c38,$0000
 dc.w    $130c,$0000
 dc.w    $2dc6,$0000
 dc.w    $18c6,$0000
 dc.w    $50e3,$0000
 dc.w    $6f1f,$0000
 dc.w    $2aaa,$0000
 dc.w    $2019,$0000
 dc.w    $50e7,$0000
 dc.w    $18c6,$0000
 dc.w    $2dce,$0000
 dc.w    $131c,$0000
 dc.w    $0c78,$0000
 dc.w    $03e0,$0000

 dc.w    $0000,$0000
 dc.w    $03e0,$0000
 dc.w    $0ff8,$0000
 dc.w    $1cfc,$0000
 dc.w    $303e,$0000
 dc.w    $203e,$0000
 dc.w    $601f,$0000
 dc.w    $70ff,$0000
 dc.w    $1111,$0000
 dc.w    $4007,$0000
 dc.w    $601f,$0000
 dc.w    $203e,$0000
 dc.w    $303e,$0000
 dc.w    $1cfc,$0000
 dc.w    $0ff8,$0000
 dc.w    $03e0,$0000

 dc.w    $0000,$0000
 dc.w    $03e0,$0000
 dc.w    $0ff8,$0000
 dc.w    $1ffc,$0000
 dc.w    $3ffe,$0000
 dc.w    $3ffe,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000

 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $3ffe,$0000
 dc.w    $3ffe,$0000
 dc.w    $1ffc,$0000
 dc.w    $0ff8,$0000
 dc.w    $03e0,$0000

 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $7fff,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000
 dc.w    $0000,$0000

 dc.w    $0000,$0000
 dc.w    $03e0,$0000
 dc.w    $0ff8,$0000
 dc.w    $1ffc,$0000
 dc.w    $3ffe,$0000
 dc.w    $3ffe,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $7fff,$0000
 dc.w    $3ffe,$0000
 dc.w    $3ffe,$0000
 dc.w    $1ffc,$0000
 dc.w    $0ff8,$0000
```

```
        dc.w    $03e0,$0000

ENEMY_A3:
        dc.w    $0000,$0000
        dc.w    $03e0,$0000
        dc.w    $0c38,$0000
        dc.w    $130c,$0000
        dc.w    $2dc6,$0000
        dc.w    $18c6,$0000
        dc.w    $50e3,$0000
        dc.w    $6f1f,$0000
        dc.w    $5555,$0000
        dc.w    $2019,$0000
        dc.w    $50e7,$0000
        dc.w    $18c6,$0000
        dc.w    $2dce,$0000
        dc.w    $131c,$0000
        dc.w    $0c78,$0000
        dc.w    $03e0,$0000

        dc.w    $0000,$0000
        dc.w    $03e0,$0000
        dc.w    $0ff8,$0000
        dc.w    $1cfc,$0000
        dc.w    $303e,$0000
        dc.w    $203e,$0000
        dc.w    $601f,$0000
        dc.w    $70ff,$0000
        dc.w    $0888,$0000
        dc.w    $4007,$0000
        dc.w    $601f,$0000
        dc.w    $203e,$0000
        dc.w    $303e,$0000
        dc.w    $1cfc,$0000
        dc.w    $0ff8,$0000
        dc.w    $03e0,$0000

        dc.w    $0000,$0000
        dc.w    $03e0,$0000
        dc.w    $0ff8,$0000
        dc.w    $1ffc,$0000
        dc.w    $3ffe,$0000
        dc.w    $3ffe,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $3ffe,$0000
        dc.w    $3ffe,$0000
        dc.w    $1ffc,$0000
        dc.w    $0ff8,$0000
        dc.w    $03e0,$0000

        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $7fff,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000

        dc.w    $0000,$0000
        dc.w    $03e0,$0000
        dc.w    $0ff8,$0000
        dc.w    $1ffc,$0000
        dc.w    $3ffe,$0000
        dc.w    $3ffe,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $3ffe,$0000
        dc.w    $3ffe,$0000
        dc.w    $1ffc,$0000
        dc.w    $0ff8,$0000
        dc.w    $03e0,$0000

ENEMY_A4:
        dc.w    $0000,$0000
        dc.w    $03e0,$0000
        dc.w    $0c38,$0000
        dc.w    $130c,$0000
        dc.w    $2dc6,$0000
        dc.w    $18c6,$0000
        dc.w    $50e3,$0000
        dc.w    $6f1f,$0000
        dc.w    $2aaa,$0000
        dc.w    $2019,$0000
        dc.w    $50e7,$0000
        dc.w    $18c6,$0000
        dc.w    $2dce,$0000
        dc.w    $131c,$0000
        dc.w    $0c78,$0000
        dc.w    $03e0,$0000

        dc.w    $0000,$0000
        dc.w    $03e0,$0000
        dc.w    $0ff8,$0000
        dc.w    $1cfc,$0000
        dc.w    $303e,$0000
        dc.w    $203e,$0000
        dc.w    $601f,$0000
        dc.w    $70ff,$0000
        dc.w    $4444,$0000
        dc.w    $4007,$0000
        dc.w    $601f,$0000
        dc.w    $203e,$0000
        dc.w    $303e,$0000
        dc.w    $1cfc,$0000
        dc.w    $0ff8,$0000
        dc.w    $03e0,$0000

        dc.w    $0000,$0000
        dc.w    $03e0,$0000
        dc.w    $0ff8,$0000
        dc.w    $1ffc,$0000
        dc.w    $3ffe,$0000
        dc.w    $3ffe,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $3ffe,$0000
        dc.w    $3ffe,$0000
        dc.w    $1ffc,$0000
        dc.w    $0ff8,$0000
        dc.w    $03e0,$0000

        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $7fff,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000

        dc.w    $0000,$0000
        dc.w    $03e0,$0000
        dc.w    $0ff8,$0000
        dc.w    $1ffc,$0000
        dc.w    $3ffe,$0000
        dc.w    $3ffe,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $7fff,$0000
        dc.w    $3ffe,$0000
        dc.w    $3ffe,$0000
        dc.w    $1ffc,$0000
        dc.w    $0ff8,$0000
        dc.w    $03e0,$0000

EXP_1:
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0100,$0000
        dc.w    $0460,$0000
        dc.w    $05d0,$0000
        dc.w    $02b0,$0000
        dc.w    $1158,$0000
        dc.w    $0d48,$0000
        dc.w    $0288,$0000
        dc.w    $05d0,$0000
        dc.w    $0200,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000

        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0100,$0000
        dc.w    $02c0,$0000
        dc.w    $0010,$0000
        dc.w    $0900,$0000
        dc.w    $1088,$0000
        dc.w    $0150,$0000
        dc.w    $1540,$0000
        dc.w    $0800,$0000
        dc.w    $0108,$0000
        dc.w    $0460,$0000
        dc.w    $0180,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000

        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0100,$0000
        dc.w    $03c0,$0000
        dc.w    $07f0,$0000
        dc.w    $0ef0,$0000
        dc.w    $1f78,$0000
        dc.w    $1ea8,$0000
```

```
        dc.w    $1ab8,$0000
        dc.w    $0ff8,$0000
        dc.w    $06f8,$0000
        dc.w    $07e0,$0000
        dc.w    $0180,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0280,$0000
        dc.w    $0420,$0000
        dc.w    $0800,$0000
        dc.w    $0008,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $1000,$0000
        dc.w    $0800,$0000
        dc.w    $0010,$0000
        dc.w    $0240,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0380,$0000
        dc.w    $07e0,$0000
        dc.w    $0ff0,$0000
        dc.w    $0ff8,$0000
        dc.w    $1ff8,$0000
        dc.w    $1ff8,$0000
        dc.w    $1ff8,$0000
        dc.w    $1ff8,$0000
        dc.w    $0ff8,$0000
        dc.w    $07f0,$0000
        dc.w    $03c0,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0380,$0000
        dc.w    $07e0,$0000
        dc.w    $0ff0,$0000
        dc.w    $0ff8,$0000
        dc.w    $1ff8,$0000
        dc.w    $1ff8,$0000
        dc.w    $1ff8,$0000
        dc.w    $1ff8,$0000
        dc.w    $0ff8,$0000
        dc.w    $07f0,$0000
        dc.w    $03c0,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
      EXP_2:
        dc.w    $0200,$0000
        dc.w    $0700,$0000
        dc.w    $0ae0,$0000
        dc.w    $1140,$0000
        dc.w    $22a8,$0000
        dc.w    $4eac,$0000
        dc.w    $2afb,$0000
        dc.w    $c3f6,$0000
        dc.w    $af5a,$0000
        dc.w    $5364,$0000
        dc.w    $0d30,$0000
        dc.w    $28a2,$0000
        dc.w    $37ca,$0000
        dc.w    $2d54,$0000
        dc.w    $10a8,$0000
        dc.w    $0510,$0000
        dc.w    $0400,$0000
        dc.w    $0880,$0000
        dc.w    $1210,$0000
        dc.w    $2108,$0000
        dc.w    $4220,$0000
        dc.w    $0e22,$0000
        dc.w    $a2e0,$0000
        dc.w    $03e5,$0000
        dc.w    $0f58,$0000
        dc.w    $9364,$0000
        dc.w    $4131,$0000
        dc.w    $0021,$0000
        dc.w    $4708,$0000
        dc.w    $4d12,$0000
        dc.w    $2044,$0000
        dc.w    $08a8,$0000
        dc.w    $0600,$0000
        dc.w    $0f80,$0000
        dc.w    $1df0,$0000
        dc.w    $3ef8,$0000
        dc.w    $7dd8,$0000
        dc.w    $71de,$0000
        dc.w    $dd1f,$0000
        dc.w    $fc1b,$0000
        dc.w    $f0a6,$0000
        dc.w    $ec9a,$0000
        dc.w    $7ecf,$0000
        dc.w    $3fdf,$0000
        dc.w    $78f6,$0000
        dc.w    $72ee,$0000
        dc.w    $3ffc,$0000
        dc.w    $0fb8,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0600,$0000
        dc.w    $0f80,$0000
        dc.w    $1ff0,$0000
        dc.w    $3ff8,$0000
        dc.w    $7ff8,$0000
        dc.w    $7ffe,$0000
        dc.w    $ffff,$0000
        dc.w    $ffff,$0000
        dc.w    $fffe,$0000
        dc.w    $fffe,$0000
        dc.w    $7fff,$0000
        dc.w    $3fff,$0000
        dc.w    $7ffe,$0000
        dc.w    $7ffe,$0000
        dc.w    $3ffc,$0000
        dc.w    $0fb8,$0000
        dc.w    $0600,$0000
        dc.w    $0f80,$0000
        dc.w    $1ff0,$0000
        dc.w    $3ff8,$0000
        dc.w    $7ff8,$0000
        dc.w    $7ffe,$0000
        dc.w    $ffff,$0000
        dc.w    $ffff,$0000
        dc.w    $fffe,$0000
        dc.w    $fffe,$0000
        dc.w    $7fff,$0000
        dc.w    $3fff,$0000
        dc.w    $7ffe,$0000
        dc.w    $7ffe,$0000
        dc.w    $3ffc,$0000
        dc.w    $0fb8,$0000
      EXP_3:
        dc.w    $0200,$0000
        dc.w    $0700,$0000
        dc.w    $0ae0,$0000
        dc.w    $1140,$0000
        dc.w    $22a8,$0000
        dc.w    $4eac,$0000
        dc.w    $295b,$0000
        dc.w    $c336,$0000
        dc.w    $ac3a,$0000
        dc.w    $5004,$0000
        dc.w    $0f70,$0000
        dc.w    $29a2,$0000
        dc.w    $37ca,$0000
        dc.w    $2d54,$0000
        dc.w    $10a8,$0000
        dc.w    $0510,$0000
        dc.w    $0400,$0000
        dc.w    $0880,$0000
        dc.w    $1210,$0000
        dc.w    $2108,$0000
        dc.w    $4220,$0000
        dc.w    $0e22,$0000
        dc.w    $a080,$0000
        dc.w    $0145,$0000
        dc.w    $0a58,$0000
        dc.w    $9244,$0000
        dc.w    $41b1,$0000
        dc.w    $0021,$0000
        dc.w    $4708,$0000
        dc.w    $4d12,$0000
        dc.w    $2044,$0000
        dc.w    $08a8,$0000
        dc.w    $0600,$0000
        dc.w    $0f80,$0000
        dc.w    $1df0,$0000
        dc.w    $3ef8,$0000
        dc.w    $7dd8,$0000
        dc.w    $71de,$0000
        dc.w    $dfff,$0000
        dc.w    $ff7b,$0000
        dc.w    $f666,$0000
        dc.w    $ee7a,$0000
        dc.w    $7fcf,$0000
        dc.w    $3fdf,$0000
        dc.w    $78f6,$0000
        dc.w    $72ee,$0000
        dc.w    $3ffc,$0000
        dc.w    $0fb8,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0600,$0000
        dc.w    $0f80,$0000
        dc.w    $1ff0,$0000
        dc.w    $3ff8,$0000
        dc.w    $7ff8,$0000
        dc.w    $7ffe,$0000
        dc.w    $ffff,$0000
        dc.w    $ff7f,$0000
        dc.w    $fe7e,$0000
        dc.w    $fe7e,$0000
        dc.w    $7fff,$0000
        dc.w    $3fff,$0000
        dc.w    $7ffe,$0000
        dc.w    $7ffe,$0000
        dc.w    $3ffc,$0000
        dc.w    $0fb8,$0000
        dc.w    $0600,$0000
        dc.w    $0f80,$0000
        dc.w    $1ff0,$0000
        dc.w    $3ff8,$0000
```

```
        dc.w    $7ff8,$0000
        dc.w    $7ffe,$0000
        dc.w    $ffff,$0000
        dc.w    $ff7f,$0000
        dc.w    $fe7e,$0000
        dc.w    $fe7e,$0000
        dc.w    $7fff,$0000
        dc.w    $3fff,$0000
        dc.w    $7ffe,$0000
        dc.w    $7ffe,$0000
        dc.w    $3ffc,$0000
        dc.w    $0fb8,$0000
EXP_4:
        dc.w    $0200,$0000
        dc.w    $0700,$0000
        dc.w    $0ae0,$0000
        dc.w    $1300,$0000
        dc.w    $2028,$0000
        dc.w    $4014,$0000
        dc.w    $300b,$0000
        dc.w    $d08a,$0000
        dc.w    $a106,$0000
        dc.w    $4288,$0000
        dc.w    $1000,$0000
        dc.w    $2c16,$0000
        dc.w    $362a,$0000
        dc.w    $2d54,$0000
        dc.w    $10a8,$0000
        dc.w    $0510,$0000
        dc.w    $0400,$0000
        dc.w    $0880,$0000
        dc.w    $1210,$0000
        dc.w    $24c8,$0000
        dc.w    $4900,$0000
        dc.w    $1422,$0000
        dc.w    $a190,$0000
        dc.w    $0a41,$0000
        dc.w    $0254,$0000
        dc.w    $9300,$0000
        dc.w    $4811,$0000
        dc.w    $0205,$0000
        dc.w    $4548,$0000
        dc.w    $4d12,$0000
        dc.w    $2044,$0000
        dc.w    $08a8,$0000
        dc.w    $0600,$0000
        dc.w    $0f80,$0000
        dc.w    $1df0,$0000
        dc.w    $3ff8,$0000
        dc.w    $7938,$0000
        dc.w    $743e,$0000
        dc.w    $d19f,$0000
        dc.w    $fbcf,$0000
        dc.w    $e3da,$0000
        dc.w    $f38e,$0000
        dc.w    $781f,$0000
        dc.w    $3e1b,$0000
        dc.w    $7b76,$0000
        dc.w    $72ee,$0000
        dc.w    $3ffc,$0000
        dc.w    $0fb8,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $00c0,$0000
        dc.w    $0040,$0000
        dc.w    $0000,$0000
        dc.w    $0400,$0000
        dc.w    $0040,$0000
        dc.w    $0180,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0600,$0000
        dc.w    $0f80,$0000
        dc.w    $1ff0,$0000
        dc.w    $3ff8,$0000
        dc.w    $7938,$0000
        dc.w    $74fe,$0000
        dc.w    $f1df,$0000
        dc.w    $fbcf,$0000
        dc.w    $e7de,$0000
        dc.w    $f3ce,$0000
        dc.w    $799f,$0000
        dc.w    $3e1f,$0000
        dc.w    $7f7e,$0000
        dc.w    $7ffe,$0000
        dc.w    $3ffc,$0000
        dc.w    $0fb8,$0000
        dc.w    $0600,$0000
        dc.w    $0f80,$0000
        dc.w    $1ff0,$0000
        dc.w    $3ff8,$0000
        dc.w    $7938,$0000
        dc.w    $74fe,$0000
        dc.w    $f1df,$0000
        dc.w    $fbcf,$0000
        dc.w    $e7de,$0000
        dc.w    $f3ce,$0000
        dc.w    $799f,$0000
        dc.w    $3e1f,$0000
        dc.w    $7f7e,$0000
        dc.w    $7ffe,$0000
        dc.w    $3ffc,$0000
        dc.w    $0fb8,$0000
EXP_5:
        dc.w    $0200,$0000
        dc.w    $0300,$0000
        dc.w    $0a40,$0000
        dc.w    $1000,$0000
        dc.w    $0008,$0000
        dc.w    $0004,$0000
        dc.w    $2009,$0000
        dc.w    $8080,$0000
        dc.w    $2006,$0000
        dc.w    $0288,$0000
        dc.w    $0000,$0000
        dc.w    $0016,$0000
        dc.w    $102a,$0000
        dc.w    $2414,$0000
        dc.w    $10a8,$0000
        dc.w    $0510,$0000
        dc.w    $0400,$0000
        dc.w    $0880,$0000
        dc.w    $0210,$0000
        dc.w    $2488,$0000
        dc.w    $4900,$0000
        dc.w    $1422,$0000
        dc.w    $a090,$0000
        dc.w    $0a41,$0000
        dc.w    $0014,$0000
        dc.w    $8300,$0000
        dc.w    $4811,$0000
        dc.w    $0205,$0000
        dc.w    $4148,$0000
        dc.w    $4412,$0000
        dc.w    $2000,$0000
        dc.w    $08a8,$0000
        dc.w    $0600,$0000
        dc.w    $0b80,$0000
        dc.w    $0d50,$0000
        dc.w    $3488,$0000
        dc.w    $4908,$0000
        dc.w    $342e,$0000
        dc.w    $8099,$0000
        dc.w    $8ac1,$0000
        dc.w    $601a,$0000
        dc.w    $a38a,$0000
        dc.w    $4815,$0000
        dc.w    $1213,$0000
        dc.w    $5962,$0000
        dc.w    $600e,$0000
        dc.w    $35a8,$0000
        dc.w    $0fb8,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $00c0,$0000
        dc.w    $0040,$0000
        dc.w    $0000,$0000
        dc.w    $0400,$0000
        dc.w    $0040,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0600,$0000
        dc.w    $0b80,$0000
        dc.w    $0f50,$0000
        dc.w    $3488,$0000
        dc.w    $4908,$0000
        dc.w    $34ee,$0000
        dc.w    $a0d9,$0000
        dc.w    $8ac1,$0000
        dc.w    $641e,$0000
        dc.w    $a3ca,$0000
        dc.w    $4815,$0000
        dc.w    $1217,$0000
        dc.w    $596a,$0000
        dc.w    $641e,$0000
        dc.w    $35a8,$0000
        dc.w    $0fb8,$0000
        dc.w    $0600,$0000
        dc.w    $0b80,$0000
        dc.w    $0f50,$0000
        dc.w    $3488,$0000
        dc.w    $4908,$0000
        dc.w    $34ee,$0000
        dc.w    $a0d9,$0000
        dc.w    $8ac1,$0000
        dc.w    $641e,$0000
        dc.w    $a3ca,$0000
        dc.w    $4815,$0000
        dc.w    $1217,$0000
        dc.w    $596a,$0000
        dc.w    $641e,$0000
        dc.w    $35a8,$0000
        dc.w    $0fb8,$0000
EXP_6:
        dc.w    $0000,$0000
        dc.w    $0300,$0000
        dc.w    $0840,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $0004,$0000
        dc.w    $0001,$0000
        dc.w    $8080,$0000
        dc.w    $0002,$0000
        dc.w    $0280,$0000
        dc.w    $0000,$0000
        dc.w    $0000,$0000
        dc.w    $1020,$0000
        dc.w    $0000,$0000
        dc.w    $10a0,$0000
        dc.w    $0010,$0000
        dc.w    $0400,$0000
        dc.w    $0800,$0000
        dc.w    $0000,$0000
        dc.w    $0088,$0000
        dc.w    $0100,$0000
        dc.w    $1402,$0000
        dc.w    $0010,$0000
        dc.w    $0840,$0000
        dc.w    $0010,$0000
        dc.w    $8200,$0000
        dc.w    $4811,$0000
        dc.w    $0201,$0000
        dc.w    $0140,$0000
        dc.w    $4000,$0000
```

```
dc.w     $0000,$0000
dc.w     $0080,$0000
dc.w     $0000,$0000
dc.w     $0400,$0000
dc.w     $0b00,$0000
dc.w     $0c40,$0000
dc.w     $0088,$0000
dc.w     $0100,$0000
dc.w     $1406,$0000
dc.w     $0011,$0000
dc.w     $88c0,$0000
dc.w     $001a,$0000
dc.w     $8280,$0000
dc.w     $4811,$0000
dc.w     $0201,$0000
dc.w     $1960,$0000
dc.w     $4000,$0000
dc.w       $15a0,$0000
dc.w     $0090,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $00c0,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0400,$0000
dc.w     $0040,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0400,$0000
dc.w     $0b00,$0000
dc.w     $0c40,$0000
dc.w     $0088,$0000
dc.w     $0100,$0000
dc.w     $14c6,$0000
dc.w     $0011,$0000
dc.w     $88c0,$0000
dc.w     $041a,$0000
dc.w     $82c0,$0000
dc.w     $4811,$0000
dc.w     $0201,$0000
dc.w     $1960,$0000
dc.w     $4000,$0000
dc.w     $15a0,$0000
dc.w     $0090,$0000
dc.w     $0000,$0000
dc.w     $0400,$0000
dc.w     $0b00,$0000
dc.w     $0c40,$0000
dc.w     $0088,$0000
dc.w     $0100,$0000
dc.w     $14c6,$0000
dc.w     $0011,$0000
dc.w     $88c0,$0000
dc.w     $041a,$0000
dc.w     $82c0,$0000
dc.w     $4811,$0000
dc.w     $0201,$0000
dc.w     $1960,$0000
dc.w     $4000,$0000
dc.w     $15a0,$0000
dc.w     $0090,$0000
dc.w     $0000,$0000
EXP_7:
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0040,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0004,$0000
dc.w     $0101,$0000
dc.w     $8000,$0000
dc.w     $0020,$0000
dc.w     $0200,$0000
dc.w     $0000,$0000
dc.w     $0020,$0000
```

```
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
         $0000,$0000
dc.w     $0400,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0088,$0000
dc.w     $0000,$0000
dc.w     $1000,$0000
dc.w     $0100,$0000
dc.w     $0800,$0000
dc.w     $0020,$0000
dc.w     $0200,$0000
dc.w     $0800,$0000
dc.w     $0020,$0000
dc.w     $0000,$0000
dc.w     $4000,$0000
dc.w     $0000,$0000
dc.w     $0080,$0000
         $0000,$0000
dc.w     $0400,$0000
dc.w     $0000,$0000
dc.w     $0440,$0000
dc.w     $0088,$0000
dc.w     $0000,$0000
dc.w     $1004,$0000
dc.w     $0101,$0000
dc.w     $8800,$0000
dc.w     $0020,$0000
dc.w     $0200,$0000
dc.w     $0800,$0000
dc.w     $0020,$0000
dc.w     $0800,$0000
dc.w     $4000,$0000
dc.w     $0000,$0000
dc.w     $0080,$0000
         $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0000,$0000
dc.w     $0400,$0000
dc.w     $0000,$0000
dc.w     $0440,$0000
dc.w     $0088,$0000
dc.w     $0000,$0000
dc.w     $1004,$0000
dc.w     $0101,$0000
dc.w     $8800,$0000
dc.w     $0020,$0000
dc.w     $0200,$0000
dc.w     $0800,$0000
dc.w     $0020,$0000
dc.w     $0800,$0000
dc.w     $4000,$0000
dc.w     $0000,$0000
dc.w     $0080,$0000
         $0000,$0000
dc.w     $0400,$0000
dc.w     $0000,$0000
dc.w     $0440,$0000
dc.w     $0088,$0000
dc.w     $0000,$0000
dc.w     $1004,$0000
dc.w     $0101,$0000
dc.w     $8800,$0000
dc.w     $0020,$0000
dc.w     $0200,$0000
dc.w     $0800,$0000
```

```
dc.w     $0020,$0000
dc.w     $0800,$0000
dc.w     $4000,$0000
dc.w     $0000,$0000
dc.w     $0080,$0000
; ---------------------------------------------
;
[...]
MSO_LOOP:
  TST.W    D3            ; Schleifenzähler testen
  BMI      TEST_BUTTONS ; Beenden, wenn negativ
  CMP.L    #MAXX_SO,0(A0,D3.W) ; mit dem
                      Maximalwert vergleichen
  BLE.S    SHOOT_X_OK   ; wenn ≤, dann OK
  SUBQ.W   #1,AKT_SO    ; ein Stern weniger
  BEQ      TEST_BUTTONS ; beenden, wenn kein
                      Schuß mehr
  MOVE.W   D3,D5        ; Zeiger kopieren
  MOVE.W   D4,D6        ; Zähler kopieren
MSO_COPY_LOOP:
  SUBQ.W   #1,D6        ; Zähler veringern
  BMI      MSO_NEXT     ; überspr. wenn negativ
  MOVE.L   4(A0,D5),0(A0,D5) ; X-Pos. kopieren
[...]
  MOVE.W   #SO_BLIT,BLTSIZE     ; Blitter Starten
  ADDI.L   #SHOOT_S,0(A0,D3.W) ; Speed zur X-
                      Position addieren
MSO_NEXT:
  SUBQ.W   #4,D3        ; Schleifenzähler veringern
  ADDQ.W   #1,D4        ; einen Schuß mehr
  BRA      MSO_LOOP     ; wenn neg, dann beenden
TEST_BUTTONS:
  BTST     #7,CIAA_PRA ; Joystickknopf gedrückt?
  SNE      J_RELEASED  ; Wenn ja, dann setzen
[...]
; ---------------------------------------------
;
; Routinen zur Verwaltung der Angreifer
;
; ---------------------------------------------
DEL_ENEMYS:
  JSR      WAIT_BLT    ; auf Ende des Blits warten
  MOVE.W   #$0100,BLTCON0 ; D auf Null gesetzt
  MOVE.W   #$0000,BLTCON1 ; 2.Kontr.Reg. setzen
  MOVE.W   #FIELD_B-ENEMY_B,BLTDMOD ; Mod setzen
  LEA      EN_BE_LAST,A0 ; Adresse der Adressen
  MOVE.W   #EN1_N-1,D2   ; Anzahl der Angreifer
DEN_LOOP2:
  MOVE.L   (A0)+,D0     ; Schiff vor 2 Bildern
  BEQ.S    DEN_NEXT     ; wenn Null:kein Angreifer
  MOVEQ    #4,D1        ; 5 Planes bearbeiten
DEN_LOOP1:
  JSR      WAIT_BLT    ; auf Ende des Blits warten
  MOVE.L   D0,BLTDPTH; Dort soll Blitter löschen
  MOVE.W   #EN_BLIT,BLTSIZE ; Blitter Starten
  ADDI.L   #FIELD_P,D0 ; eine Plane weiter gehen
  DBRA     D1,DEN_LOOP1     ; nächster Blit
DEN_NEXT:
  DBRA     D2,DEN_LOOP2     ; nächster Angreifer
DEL_EN_END:
  LEA      EN_LAST,A0   ; Adresse der Adressen
  LEA      EN_BE_LAST,A1  ; dito
  MOVE.W   #EN1_N-1,D0  ; Anzahl der Angreifer
DEN_LOOP3:
  MOVE.L   (A0)+,(A1)+  ; vorletzte wird letzte
  DBRA     D0,DEN_LOOP3 ; Schleife
DEN_END:
  RTS
MOVE_ENEMYS:
  JSR      INIT_ENEMYS1 ; --> Bewegungroutine
  LEA      EN_ANIM,A0   ; Animationsphasen
```

```
 LEA      EN_X,A1     ; Adresse der X-Werte nach A0
 LEA      EN_Y,A2     ; Adresse der Y-Werte nach A1
 LEA      EN_LAST,A3  ; Adresse der Zw.-speicher
 LEA      EN_GFX,A4   ; Adresse der Bobadressen
 MOVEQ    #0,D3       ; Long-Offset
 MOVEQ    #0,D4       ; Byte-Offset
 MOVEQ    #EN1_N-1,D5 ; Zähler
MEN_LOOP:
 TST.B    (A0,D4)     ; Nicht vorhanden?
 BEQ      MEN_OFF     ; nächsten Teil überspringen

 MOVE.L   (A1,D3),D0  ; D0 mit X-Position laden
 MOVE.L   (A2,D3),D1  ; D1 mit Y-Position laden
 JSR      CALC_POS    ; Position umrechnen

 LSL.W    #8,D1 ; Shiftwert um 8 Bits nach rechts
 LSL.W    #4,D1 ; Shiftwert um 4 Bits nach rechts
 JSR      WAIT_BLT    ; auf Ende des Blits warten
 MOVE.W   D1,BLTCON1  ; zweites Kontr.-reg. setzen
 MOVE.W   D1,D7       ; D1 zwischenspeichern
 ADDI.W   #$0fca,D1   ; ABCD an und D=AB+Ac
 MOVE.W   D1,BLTCON0  ; erstes Kontrollreg. setzen

 CLR.W    BLTAMOD ; Schattenmaske: kein Modulo
 CLR.W    BLTBMOD ; Angreiferdaten: kein Modulo
 MOVE.W   #FIELD_B-ENEMY_B,BLTCMOD ; Mod. setzen
 MOVE.W   #FIELD_B-ENEMY_B,BLTDMOD ; Mod. setzen
 ADD.L    FIELD_BASE1,D0  ; Spielfeldadr. zu D0
 ADD.L    D2,D0          ; Y-Offset dazu
 MOVE.L   D0,(A3,D3)     ; neue Adr. abspeichern

 MOVEQ    #0,D2          ; D2 löschen
 MOVE.B   (A0,D4),D2     ; Animationsphase holen
 LSL.L    #2,D2        ; auf Longwordzeiger bringen
 MOVE.L   (A4,D2),D2 ; Adr. der Grafikdaten holen
 MOVE.L   D2,BLTBPTH ; Zeiger auf Angreiferdaten
 ADDI.L   #5*ENEMY_P,D2  ; Adr. der Schattenmaske

 MOVEQ    #4,D1  ; 5 Planes bearbeiten
MEN_LOOP2:
 JSR      WAIT_BLT   ; auf Ende des Blits warten
 MOVE.L   D2,BLTAPTH ; Zeiger auf Schattenmaske
 MOVE.L   D0,BLTCPTH ; Dort liegt der Hintergrund
 MOVE.L   D0,BLTDPTH ; Dort soll Blitter hinkop.
 MOVE.W   #EN_BLIT,BLTSIZE   ; Blitter starten
 ADDI.L   #FIELD_P,D0    ; eine Plane weitergehen
 DBRA     D1,MEN_LOOP2   ; nächster Blit

 CMP.B    #EXP_ST,(A0,D4) ; Explodiert Angreifer?
 BGE.S    MEN_NEXT ; dann keine Kollisionsabfrage

 JSR      WAIT_BLT    ; auf Ende des Blits warten
 MOVE.W   D7,BLTCON1  ; zweites Kontr.-reg setzen
 MOVE.W   #$0cc0,BLTCON0   ; 1.Kontr.-Reg. setzen
 MOVE.W   #FIELD_B-ENEMY_B,BLTAMOD ; Bildschirm-
                        daten: Modulo setzen
 MOVE.L   D0,BLTAPTH      ; Schiffskollisionsplane
 MOVE.W   #EN_BLIT,BLTSIZE   ; Blitter starten
 JSR      WAIT_BLT    ; auf Ende des Blits warten
 MOVE.W   DMACONR,D6  ; DMA-Register auslesen
 BTST     #13,D6      ; Blitter-Zero-Bit testen
 BNE.S    NOT_DEAD    ; wenn auf 1,nicht getroffen
 MOVE.L   #START_OVER,MAIN_JMP+2  ; Vektor der
                        Hauptschleife verbiegen
NOT_DEAD:
 ADDI.L   #FIELD_P,D0   ; eine Plane weiter gehen
 MOVE.L   D0,BLTAPTH    ; Schiffskollisionsplane
 MOVE.L   D2,BLTBPTH    ; Zeiger auf Schatten-
                          maske setzen
 MOVE.W   #EN_BLIT,BLTSIZE   ; Blitter starten
 JSR      WAIT_BLT    ; auf Ende des Blits warten
 MOVE.W   DMACONR,D6  ; DMA-Register auslesen
 BTST     #13,D6      ; Blitter-Zero-Bit testen
 BNE.S    MEN_NEXT  ; wenn auf 1, nicht getroffen
 MOVE.B   #EXP_ST-1,(A0,D4)  ; Angreifer aus
MEN_NEXT:
 ADDQ.W   #4,D3          ; Long-Offset erhöhen
 ADDQ.W   #1,D4          ; Byte-Offset erhöhen
 DBRA     D5,MEN_LOOP    ; wenn neg, dann beenden
MEN_END:
 RTS
MEN_OFF:
 CLR.L    (A3,D3)        ; Angreifer ausschalten
 BRA.S    MEN_NEXT       ; weiter machen

EN1_N     EQU     12     ; Anzahl der Angreifer
EN1_P     EQU     4      ; Anz. der Animationsphasen

INIT_ENEMYS1:
 MOVEQ    #EN_M-1,D0       ; Anzahl der Angreifer
 LEA      EN_ANIM,A0   ; Adr. der Animationsphasen
INIT_E1_LOOP1:
 CLR.B    (A0)+           ; Angreifer löschen
 DBRA     D0,INIT_E1_LOOP1   ; Schleife

 MOVEQ    #EN1_N-1,D1      ; Anzahl der Angreifer
 LEA      EN_1,A0     ; Adr. der Geschwindigkeiten
INIT_E1_LOOP2:
 MOVEQ    #0,D0            ; D0 löschen
 JSR      FAST_RAND        ; Zufallszahl holen
 ANDI.B   #$3,D0           ; Zahlen zwischen 0 und 3
 ADDQ.B   #1,D0            ; Zahlen zwischen 1 und 4
 MOVE.L   D0,(A0)+         ; Angreifer löschen
 DBRA     D1,INIT_E1_LOOP2    ; Schleife

 MOVE.W   #EN1_N,EN_AKT  ; Anz. der Angreifer absp.

 MOVEQ    #0,D0            ; Angreifer am Schirm
 MOVEQ    #25,D1           ; Wartezähler
 MOVEQ    #0,D2            ; Byte-Offset
 MOVEQ    #0,D3            ; Long-Offset
 MOVEQ    #0,D4            ; Byte-Offset
 MOVEQ    #0,D5            ; Long-Offset
 MOVEQ    #EN1_N-1,D6      ; Zähler setzen

 LEA      EN_X,A0          ; Adresse der X-Werte
 LEA      EN_Y,A1          ; Adresse der Y-Werte
 LEA      EN_BE_LAST,A2  ; Adr. vor 2 Videobildern
 LEA      EN_LAST,A3      ; Adr. vor einem Videobild
 LEA      EN_ANIM,A4    ; Adr. der Animationsphasen
 LEA      EN_1,A5       ; Adresse des 1. Hilfsarays
 LEA      EN_MOVEM,A6     ; Adr. des Zw.-speichers
 MOVEM.L  D0-D3,-(A6)     ; Register zw.-speichern
 MOVEM.L  D4-D6/A0-A5,-(A6) ; Reg. zw.-speichern
 MOVE.L   A6,EN_STACK     ; Adr. des Pseudostacks
 MOVE.L   #MOVE_ENEMYS1,MOVE_ENEMYS+2 ; Vektor
                                       verbiegen

MOVE_ENEMYS1:
 TST.W    EN_AKT          ; alle Angreifer erledigt?
 BNE.S    ME1_NOT_FINISHED ; nicht, weitermachen
 MOVE.L   #INIT_ENEMYS2,MOVE_ENEMYS+2 ; Vektor
                                       verbiegen
ME1_NOT_FINISHED:
 MOVEA.L  EN_STACK,A6     ; Adr. des Zw.-speichers
 MOVEM.L  (A6)+,D4-D6/A0-A5     ; Register holen
 MOVEM.L  (A6)+,D0-D3           ; Register holen
 SUBQ.W   #1,D1           ; Wartezähler erniedrigen
 BPL.S    ME1_LOOP ; noch nicht -1, dann springen
 CMP.W    #EN1_N,D0 ; schon alle Angr. am Schirm?
 BEQ.S    ME1_LOOP  ; dann überspringen
 ADDQ.W   #1,D0     ; Anzahl der Angreifer erhöhen
 MOVE.L   #367,(A0,D3) ; Anfangs X-Wert
                         schreiben 367
 MOVE.L   #303,(A1,D3) ; Anfangs Y-Wert schreiben
 MOVE.L   #0,(A2,D3)   ; Dummy-Adresse schreiben
 MOVE.B   #1,(A4,D2)   ; Animationsphase schreiben
 MOVE.W   #7,D1           ; Wartezähler neu setzen
 ADDQ.W   #1,D2           ; Byte-Offset erhöhen
 ADDQ.W   #4,D3           ; Long-Offset erhöhen
ME1_LOOP:
 TST.B    (A4,D4)       ; wird Bob nicht angezeigt?
 BEQ.S    ME1_NOT_NEG2      ; dann überspringen
 CMP.B    #EN1_P,(A4,D4)    ; letzte Phase?
 BEQ.S    ME1_LAST_ANIM     ; dann überspringen
 CMP.B    #EXP_ED,(A4,D4) ; letzte Expl.-Phase?
 BEQ.S    ME1_IS_DEAD   ; dann endlich ausschalten
 ADDQ.B   #1,(A4,D4)    ; Animationsphase erhöhen
 BRA.S    ME1_IS_ANIMATED ; nächsten Teil überspr.
ME1_IS_DEAD:
 CLR.B    (A4,D4)       ; Animationsphase auf Null
 SUBQ.W   #1,EN_AKT     ; ein Angreifer weniger
 BRA.S    ME1_IS_ANIMATED ; nächsten Teil überspr.
ME1_LAST_ANIM:
 MOVE.B   #1,(A4,D4)    ; auf erste Anim.-Phase
                          stellen
ME1_IS_ANIMATED:
 MOVE.L   (A5,D5),D7    ; Geschwindigkeit holen
```

```
 SUB.L   D7,(A0,D5)    ; X-Position erniedrigen
 BPL.S   ME1_NOT_NEG1 ; wenn positiv, springen
 CLR.B   (A4,D4)       ; den Angreifer rauswerfen
 CLR.L   (A3,D5)       ; Dummyadresse schreiben
 SUBQ.W  #1,EN_AKT     ; ein Angreifer weniger
ME1_NOT_NEG1:
 SUBQ.L  #4,(A1,D5)    ; Y-Position erniedrigen
 BPL.S   ME1_NOT_NEG2 ; wenn positiv, springen
 CLR.B   (A4,D4)       ; den Angreifer rauswerfen
 CLR.L   (A3,D5)       ; Dummyadresse schreiben
 SUBQ.W  #1,EN_AKT     ; ein Angreifer weniger
ME1_NOT_NEG2:
 ADDQ.W  #1,D4         ; Byte-Offset erhöhen
 ADDQ.W  #4,D5         ; Long-Offset erhöhen
 DBRA    D6,ME1_LOOP   ; Schleife machen
ME1_END:
 MOVEM.L D0-D3,-(A6)   ; Register zw.-speichern
 RTS

EN2_N    EQU      9    ; Anzahl der Angreifer
EN2_P    EQU      4    ; Anz. der Animationsphasen
EN2_O    EQU      5    ; Tabellenoffset zwischen
                         2 Angreifern

INIT_ENEMYS2:
 MOVEQ   #EN_M-1,D0    ; Anzahl der Angreifer
 LEA     EN_ANIM,A0    ; Adr. der Animationsphasen
INIT_E2_LOOP1:
 CLR.B   (A0)+         ; Angreifer löschen
 DBRA    D0,INIT_E2_LOOP1   ; Schleife

 MOVE.W  #EN2_N,EN_AKT  ; Anz. der Angreifer sp.

 MOVEQ   #0,D0            ; Angreifer am Schirm
 MOVEQ   #0,D1            ; Wartezähler
 MOVEQ   #0,D2            ; Byte-Offset
 MOVEQ   #0,D3            ; Long-Offset
 MOVEQ   #0,D4            ; Byte-Offset
 MOVEQ   #0,D5            ; Long-Offset
 MOVEQ   #EN2_N-1,D6      ; Zähler setzen
 MOVEQ   #0,D7            ; Tabellenoffset

 LEA     EN_X,A0          ; Adresse der X-Werte
 LEA     EN_Y,A1          ; Adresse der Y-Werte
 LEA     EN_BE_LAST,A2  ; Adr. vor 2 Videobildern
 LEA     EN_Y2,A3     ; Adr. vor einem Videobild
 LEA     EN_ANIM,A4   ; Adr. der Animationsphasen
 LEA     EN_X2,A5     ; Adr. der Bewegungstabelle
 LEA     EN_MOVEM,A6 ; Adr. des Zw.-speichers
 MOVEM.L D0-D3/D7,-(A6)      ; Reg. zw.-speichern
 MOVEM.L D4-D6/A0-A5,-(A6)  ; Reg. zw.-speichern
 MOVE.L  A6,EN_STACK ; Adr. des Pseudostacks
                       ablegen
 MOVE.L  #MOVE_ENEMYS2,MOVE_ENEMYS+2 ; Vektor
                                     verbiegen

MOVE_ENEMYS2:
 TST.W   EN_AKT        ; alle Angreifer erledigt?
 BNE.S   ME2_NOT_FINISHED  ; nicht, weitermachen
 MOVE.L  #INIT_ENEMYS1,MOVE_ENEMYS+2 ; Vektor
                                     verbiegen
ME2_NOT_FINISHED:
 MOVEA.L EN_STACK,A6      ; Adr. des Zw.-speichers
 MOVEM.L (A6)+,D4-D6/A0-A5      ; Register holen
 MOVEM.L (A6)+,D0-D3/D7         ; Register holen
 SUBQ.W  #1,D1         ; Wartezähler erniedrigen
 BPL.S   ME2_LOOP      ; noch nicht -1, springen
 CMP.W   #EN2_N,D0     ; alle Angreifer am Schirm?
 BEQ.S   ME2_LOOP      ; dann überspringen
 ADDQ.W  #1,D0         ; Anz. der Angreifer erhöhen
 MOVE.L  #0,(A2,D3) ; Dummy-Adresse schreiben
 MOVE.B  #1,(A4,D2) ; Animationsphase schreiben
 MOVE.W  #EN2_O-1,D1; Wartezähler neu setzen
 ADDQ.W  #1,D2         ; Byte-Offset erhöhen
 ADDQ.W  #4,D3         ; Long-Offset erhöhen
ME2_LOOP:
 MOVE.L  (A5,D7.L),(A0,D5); X-Pos. schreiben
 MOVE.L  (A3,D7.L),(A1,D5); Y-Pos. schreiben
 SUBA.L  #EN2_O*4,A5        ; einen Eintrag weiter
 SUBA.L  #EN2_O*4,A3        ; einen Eintrag weiter
 TST.B   (A4,D4)        ; wird Bob nicht angezeigt?
 BEQ.S   ME2_IS_ANIMATED  ; dann überspringen
 CMP.B   #EN1_P,(A4,D4) ; letzte Animationsphase?
 BEQ.S   ME2_LAST_ANIM    ; dann überspringen
 CMP.B   #EXP_ED,(A4,D4) ; letzte Expl.-Phase?
 BEQ.S   ME2_IS_DEAD       ; endlich ausschalten
 ADDQ.B  #1,(A4,D4)        ; sonst Animationsphase
                             erhöhen
 BRA.S   ME2_IS_ANIMATED ; nächsten Teil überspr.
ME2_IS_DEAD:
 CLR.B   (A4,D4)           ; Animationsphase auf 0
 SUBQ.W  #1,EN_AKT         ; ein Angreifer weniger
 BRA.S   ME2_IS_ANIMATED ; nächsten Teil überspr.
ME2_LAST_ANIM:
 MOVE.B  #1,(A4,D4)    ; auf erste Animations-
                         phase stellen
ME2_IS_ANIMATED:
 ADDQ.W  #1,D4          ; Byte-Offset erhöhen
 ADDQ.W  #4,D5          ; Long-Offset erhöhen
 DBRA    D6,ME2_LOOP  ; Schleife machen
ME2_END:
 ADDQ.L  #4,D7          ; einen Tabellenwert
                          weiter gehen
 CMP.L   #4*(EN_TL1+EN2_N*EN2_O),D7 ; Tabellen-
                                ende erreicht?
 BLE.S   ME2_NOT_TAB_END  ; wenn nicht, nächsten
                            Teil überspringen
 MOVE.L  #INIT_ENEMYS1,MOVE_ENEMYS+2 ; Vektor
                                     verbiegen
ME2_NOT_TAB_END:
 MOVEM.L D0-D3/D7,-(A6)    ; Reg. zw.-speichern
 RTS

EN_1:           DS.L    EN_M   ; Hilfsarray #1
EN_X:           DS.L    EN_M   ; X-Positionen
                                 der Schüsse
EN_Y:           DS.L    EN_M   ; Y-Positionen
                                 der Schüsse
EN_LAST: DS.L   EN_M           ; Adr. vor einem
                                 Videobild
EN_BE_LAST:     DS.L    EN_M   ; Adr. vor zwei
                                 Videobildern
         DS.L   15        ; Platz zum zw.-speichern
EN_MOVEM:                 ; Endadr. des Speichers
EN_STACK:       DS.L    1   ; Adr. des Stacks
EN_GFX:         DC.L    0   ; kein Angreifer
         DC.L   6*SHIP_P+6*SHOOT_P
         DC.L   6*SHIP_P+6*SHOOT_P+6*ENEMY_P
         DC.L   6*SHIP_P+6*SHOOT_P+2*6*ENEMY_P
         DC.L   6*SHIP_P+6*SHOOT_P+3*6*ENEMY_P
         DC.L   0
         DC.L   6*SHIP_P+6*SHOOT_P+4*6*ENEMY_P
         DC.L   6*SHIP_P+6*SHOOT_P+5*6*ENEMY_P
         DC.L   6*SHIP_P+6*SHOOT_P+6*6*ENEMY_P
         DC.L   6*SHIP_P+6*SHOOT_P+7*6*ENEMY_P
         DC.L   6*SHIP_P+6*SHOOT_P+8*6*ENEMY_P
         DC.L   6*SHIP_P+6*SHOOT_P+9*6*ENEMY_P
         DC.L   6*SHIP_P+6*SHOOT_P+10*6*ENEMY_P
EN_AKT:         DS.W    1   ; Zähler der noch
                              aktiven Angreifer
EN_ANIM: DS.B   EN_M        ; Zähler für die
                              Animationsphasen
 EVEN
EN_TL1   EQU    212         ; Länge einer Tabelle
EN_X2:
 DC.L    0,4,8,12,16,20,24,28,32,36
 DC.L    40,44,48,52,56,60,64,68,72,76
 DC.L    80,84,88,92,96,100,104,108,112,116
 DC.L    120,124,128,132,136,140,144,148,152,156
 DC.L    160,164,168,172,176,180,184,188,192,201
 DC.L    210,219,228,236,244,252,258,264,269,274
 DC.L    277,280,281,282,281,280,277,274,269,264
 DC.L    258,252,244,236,228,219,210,201,192,183
 DC.L    174,165,156,147,140,132,126,120,115,110
 DC.L    107,104,103,102,103,104,107,110,115,120
 DC.L    126,132,140,148,156,165,174,183,192,201
 DC.L    210,219,228,236,244,252,258,264,269,274
 DC.L    277,280,281,282,281,280,277,274,269,264
 DC.L    258,252,244,236,228,219,210,201,192,183
 DC.L    174,165,156,147,140,132,126,120,115,110
 DC.L    107,104,103,102,103,104,107,110,115,120
 DC.L    126,132,140,148,156,165,174,183,192,196
 DC.L    200,204,208,212,216,220,224,228,232,236
 DC.L    240,244,248,252,256,260,264,268,272,276
 DC.L    280,284,288,292,296,300,304,308,312,316
 DC.L    320,324,328,332,336,340,344,348,352,356
 DC.L    360,364
 DS.L    EN2_N*EN2_O
```

```
EN_Y2:
 DC.L     50,53,56,59,63,66,69,72,75,79
 DC.L     82,85,88,91,94,97,99,102,105,108
 DC.L     110,113,115,118,120,122,125,127,129,131
 DC.L     133,134,136,138,139,141,142,143,144,145
 DC.L     146,147,148,148,149,149,149,149,150,153
 DC.L     156,159,162,165,168,171,174,177,180,183
 DC.L     186,189,192,195,198,201,204,207,210,213
 DC.L     216,219,222,225,228,231,234,237,240,237
 DC.L     234,231,228,225,222,219,216,213,210,207
 DC.L     204,201,198,195,192,189,186,183,180,177
 DC.L     174,171,168,165,162,159,156,153,150,153
 DC.L     156,159,162,165,168,171,174,177,180,183
 DC.L     186,189,192,195,198,201,204,207,210,213
 DC.L     216,219,222,225,228,231,234,237,240,237
 DC.L     234,231,228,225,222,219,216,213,210,207
 DC.L     204,201,198,195,192,189,186,183,180,177
 DC.L     174,171,168,165,162,159,156,153,150,151
 DC.L     151,151,151,151,152,152,153,153,154,154
 DC.L     155,156,157,158,159,160,161,162,163,165
 DC.L     166,167,169,170,171,173,174,176,178,179
 DC.L     181,183,184,186,188,190,191,193,195,197
 DC.L     199,200
 DS.L     EN2_N*EN2_O


; ---------------------------------------------
;
; Ab hier stehen der Stack, unsere Variablen und Zwischenspeicher (Buffer)
;
; ---------------------------------------------

[...]
```

ENDE

KICKUP

Die Grafik

Angreifer und Explosionen

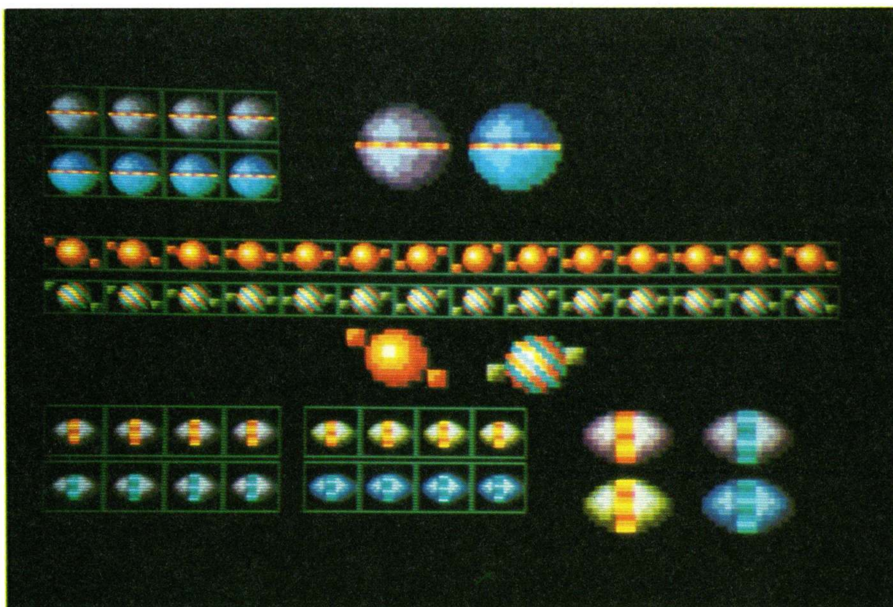
Nachdem ich Ihnen in der vorigen Ausgabe dieses Spiele-Grafik-Kurses die Animation mit Hilfe der Cycle-Funktion vorgestellt habe, sind heute die Animationen, die für unser Kickstart-Spiel verwendet werden, an der Reihe. Die größten Unterschiede dieser Verfahren sind:

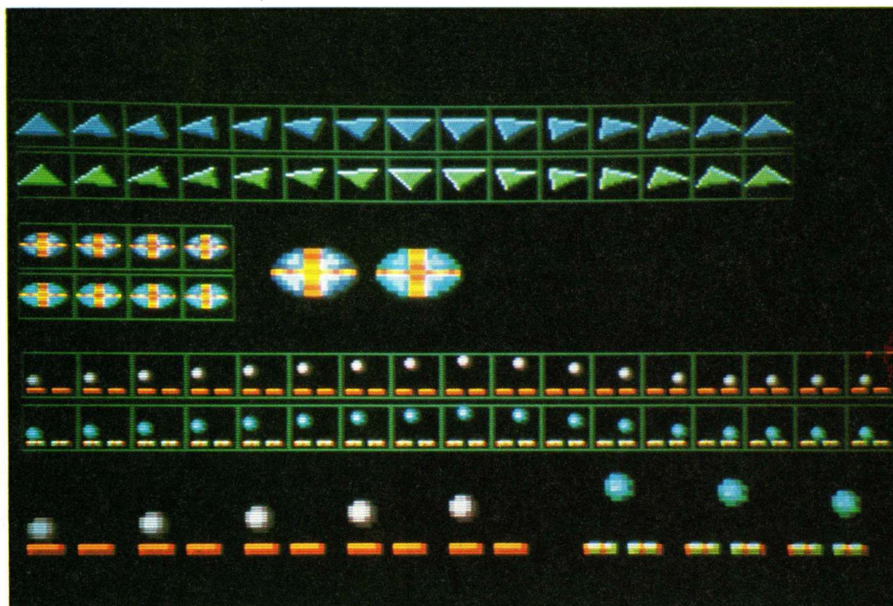
- die Anzahl der einzelnen Animationssequenzen, die bei der Cycle-Funktion durch die Breite oder Höhe des Bildschirmes begrenzt werden.

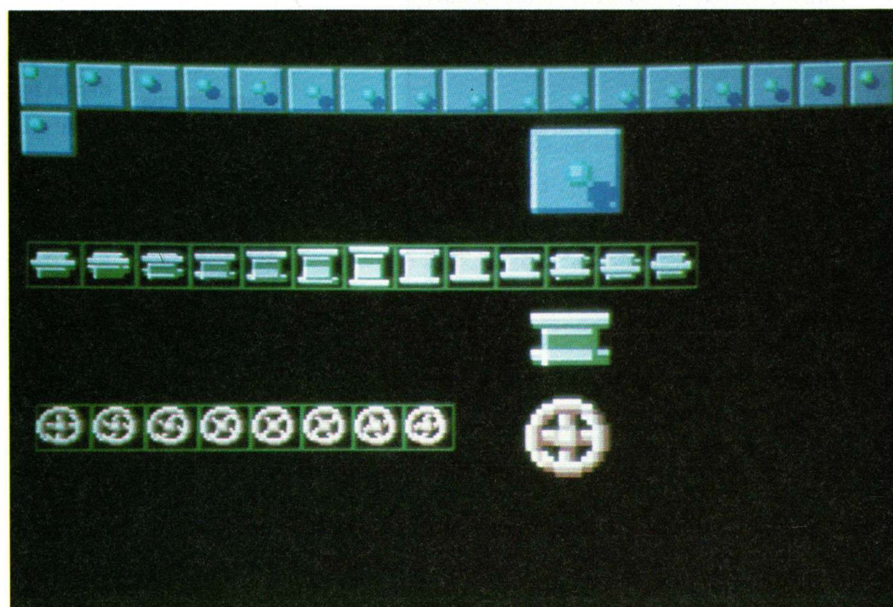
- die Anzahl der Farben. Bei der neuen Technik können Sie im Gegensatz zu der alten Methode, bei der Sie höchstens 4 Farben benutzen konnten, bis zu 31 Farben verwenden. (Eine Farbe muß als Hintergrundfarbe verwendet werden.)

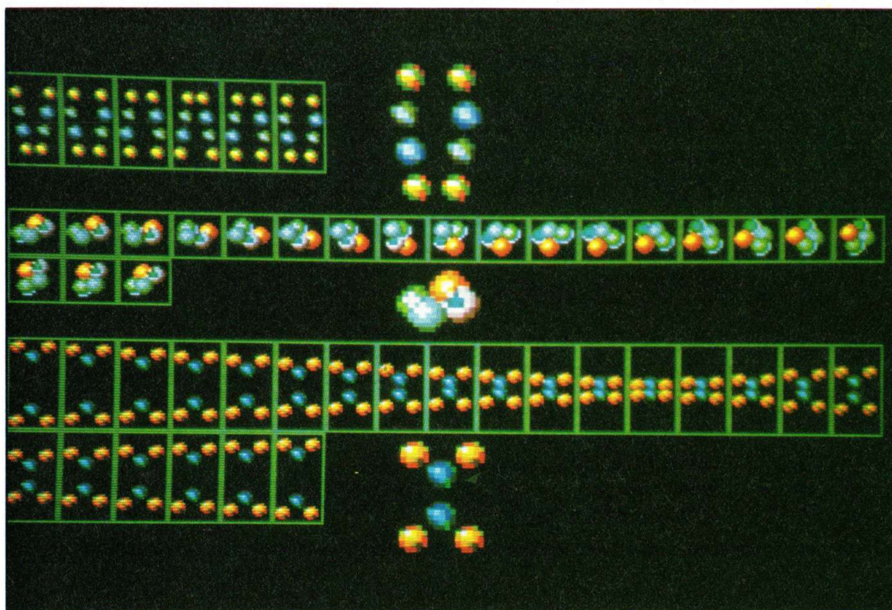
- die Möglichkeit einer viel feineren Abstufung zwischen den einzelnen Sequenzen. Bei der Animation mit dem Cycle-Modus können Sie die Animationssequenzen nur nebeneinander darstellen, wodurch große Zwischenräume zwischen den Sequenzen entstehen. Dieses Manko ist hier beseitigt. Sie können die Animation sogar auf der Stelle ablaufen lassen.

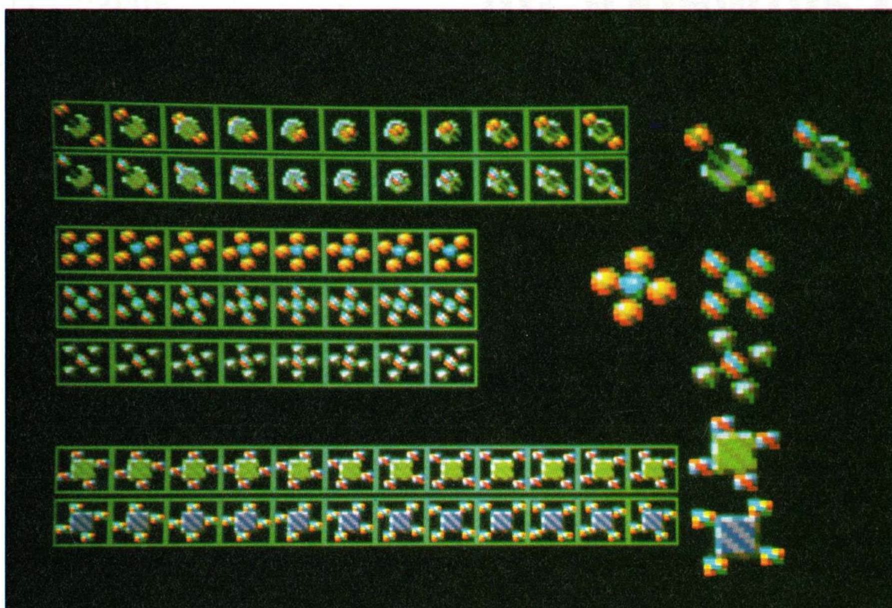
"Bei so vielen Vorteilen muß es doch auch noch einen Nachteil geben", werden Sie sich nun sicher denken. Ja, den gibt es, aber er ist nicht sehr gravierend: Sie benötigen ein Animationsprogramm, um diese Animationen bewegen zu können. Hier gibt es eine große Anzahl an Programmen, die Ihren Geldbeutel auch nicht zu sehr belasten, darunter sogar einige Public Domain-Animationsprogramme. Übrigens, wenn Sie die Routinen zur Animation der einzelnen Objekte verwenden, könnten Sie sich mit etwas Geschick solch ein Animationsprogramm auch selbst schreiben, oder?!
Aber kommen wir jetzt zu dem neuen Verfahren:
Es wird eigentlich von fast jedem herkömmlichen Spiel auf dem AMIGA verwendet. Die einzelnen Animationssequenzen werden nacheinander in einer bestimmten Reihenfolge abgespielt.

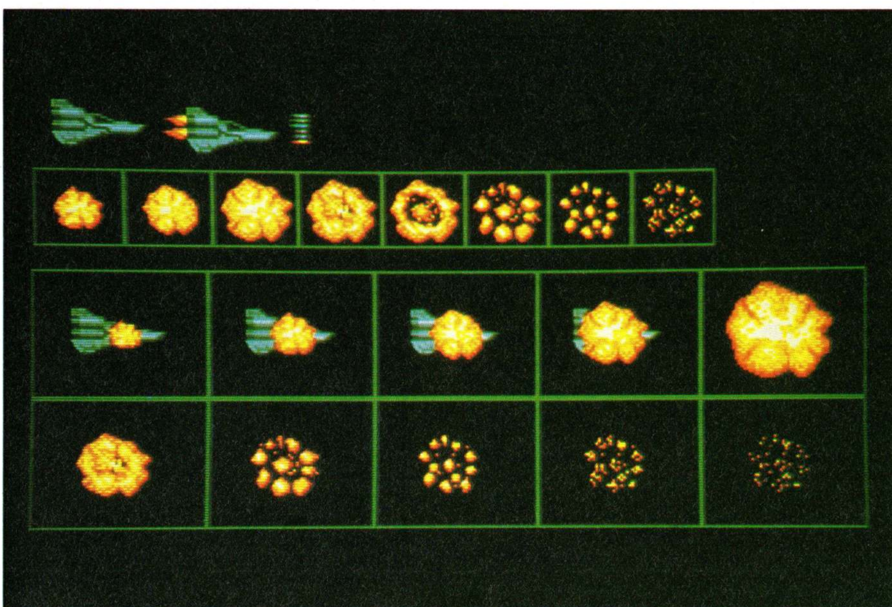












Dabei müssen Sie auf folgendes achten:

- Die Größe der Objekte sollte immer gleich bleiben, wobei aber trotzdem die Möglichkeit der Vergrößerung bzw. Verkleinerung besteht. Hilfreich ist es, einen Rand um das Objekt herumzulegen und diesen als Begrenzung zu benutzen. Mit Hilfe der Brush-Funktion sollten Sie diesen Rand mehrere Male nebeneinander auf dem Bildschirm plazieren.

- Ganz wichtig ist, daß die einzelnen Animationssequenzen auch hintereinander passen. Das heißt, Sie müssen ständig überprüfen, wie die Animation aussieht. Hierfür ist die Benutzung des Animationsprogrammes notwendig.

Wenn Sie sich ganz sicher sind, daß die einzelnen Sequenzen aneinanderpassen, müssen diese nur noch in das Spiel eingebaut werden, und die große Ballerei kann beginnen. Viel Spaß beim Zeichnen der Angreifer und anderer Objekte wünscht Ihnen

Ihr Sebastian Dosch

VON CHRISTIAN KELLER

SUPERBACK

wirklich super?

Seit kurzem ist ein neues Backupprogramm auf demMarkt erhältlich. Ob es seinem Namen, der einiges verspricht, gerecht wird, soll einmal geprüft werden.

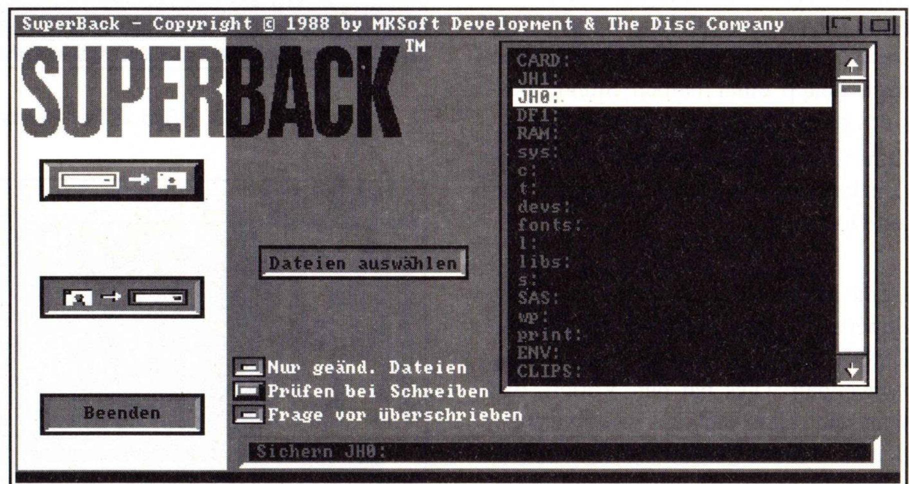


Bild 1: Alle Divices stehen beim Sichern zur Auswahl

Superback von MKSoft ist der Nachfolger von LVBackup, einem Festplattenbackupprogramm, das wir im Juliheft 88 schon getestet haben. Es mußte sich damals mit Quarterback messen, das durch seine Möglichkeiten und Geschwindigkeit bestiсht. Der Vergleich LVBackup-Quarterback ging eindeutig zugunsten von Quarterback aus. So muß sich auch Superback an den Maßstäben messen lassen, die Quarterback gesetzt hat.

Bedienungskomfort

Gegenüber seinem Vorgänger hat die Bequemlichkeit der Benutzung um einiges zugenommen. Trotzdem sind die Möglichkeiten, die das Programm bietet, nicht sonderlich vielfältig. Auf der Diskette findet man drei Versionen des Programmes, welche jeweils eine andere Sprache in der Benutzerführung verwenden. Diese Sprachen sind Deutsch, Französisch und Englisch.

Was bequem vonstatten geht, ist die Selektion des zu sichernden Devices. In einem Fenster erscheinen die Namen aller Geräte, die dem AMIGA bekannt sind, und man kann durch Anklicken das zu sichernde auswählen. Hat man das Device bestimmt und Sichern angewählt, wechselt der Bildschirm, und es wird in zwei Fenstern der Platteninhalt angezeigt. Ein Fenster enthält alle Verzeichnisse und Unterverzeichnisse, das andere zeigt den Inhalt des gerade angewählten Directories an. Auswahlmöglichkeiten bestehen nur für einzelne Dateien und ganze Verzeichnisse. Hat man seine Selektion abgeschlossen, beginnt das Backup. Superback zeigt mittels Requester das Einlegen neuer Disketten an. Zur Sicherung wird nur das interne Laufwerk benutzt.

Beim Sichern werden verschiedene Statusinformationen angezeigt. Dazu gehören neben der belegten Puffergröße, der Diskettennummer und dem verbleibenden Platz auf dieser auch die noch zu sichernden Dateien und deren Gesamtgröße. Diese Gesamtgröße ließ mich aber etwas erstaunt blicken, zeigt sie doch an, daß noch 1157 Megabyte zu sichern seien, obwohl nur 20 Mbyte Platz auf der Platte sind. Auch die Angabe der Dateizahl liegt in Bereichen, die nicht mit der Anzahl der vorhandenen Dateien übereinstimmen kann. So nett solche Zusatzinformationen auch sind, richtig sollten sie schon sein. Ansonsten läßt man sie besser ganz weg. Zudem zeigt Superback am Ende immer eine nicht gesicherte Datei an, was nicht gerade ein beruhigendes Gefühl hinterläßt.

Die Einflußname auf den Backupvorgang selbst beschränkt sich auf drei Punkte. Der erste ist die Beachtung des Archiveflags. Mit dieser Option werden nur die Dateien, bei denen dieses Flag gesetzt ist, gesichert. Dadurch ist die additive Sicherung möglich. Leider schreibt Superback das Archiveflag immer auf Null. Eine Möglichkeit, es in seinem Zustand zu belassen, wie sie Quarterback bietet, ist nicht gegeben.
Eine weitere Option ermöglicht es, Disketten vor dem Schreiben auf DOS-Format zu prüfen und gegebenenfalls eine Sicherheitsabfrage zu tätigen, bevor die Diskette überschrieben wird. Superback kann die Disketten auch überprüfen,

was mit LVBackup nicht möglich war. Die Datensicherheit wird dadurch erheblich erhöht, was den Geschwindigkeitsverlust mehr als wettmacht.

Was Superback in keinem guten Licht erscheinen läßt ist sein Verhalten beim Auftreten eines Diskettenfehlers beim Sichern. Anstatt den Benutzer darauf hinzuweisen und eine neue Diskette zu verlangen, zeigt Superback dem armen Menschen den Fehler an und -hört mit dem Backup auf. Der Grund, ein derartiges Verhalten in das Programm einzubauen, weiß wahrscheinlich der Programmierer ganz alleine. Wenn einem sowas bei der 12ten von 14 Disketten nach knapp einer halben Stunde passiert, ja dann könnte man diesen Herrn ...

Zum Glück verhält sich Superback anders bei versehentlich schreibgeschützten Disketten.

Gemeinsamkeiten mit dem Vorgänger Superback kann Disketten, die mit LVBackup 5.1a geschrieben wurden, verwenden. Dies hat den Vorteil, daß man seine alten Backups weiterverwenden kann. Benutzt man keine Komplettsicherung, sondern arbeitet mit sogenannten Additivsicherungen, bei denen nur die veränderten Dateien gesichert werden, kann man so seine alten Backups weiterbenutzen.

Geschwindigkeit

Die Zeit, die benötigt wird, um ein Backup durchzuführen, ist entscheidend für die Häufigkeit, mit der man diese dann auch durchführt. Daher ist der Geschwindigkeit ein besonderes Augenmerk zu widmen. Auch hier wurde zum Vergleich Quarterback herangezogen.

Die Versuche fanden mit einer Janus-Festplatte statt, die mit 13 MByte Daten gefüllt war. Mit der Verify-Option sah der Vergleich wie folgt aus:

Superback:	30 Minuten
Quarterback:	23 Minuten

Der Sieger steht also eindeutig fest. Ohne Verify legt Superback zwar erheblich zu, mehr als Quarterback gewinnt, diese Art des Backups wird aber vom Autor abgelehnt. Daher wurden auch keine genauen Messungen ohne Überprüfung durchgeführt.

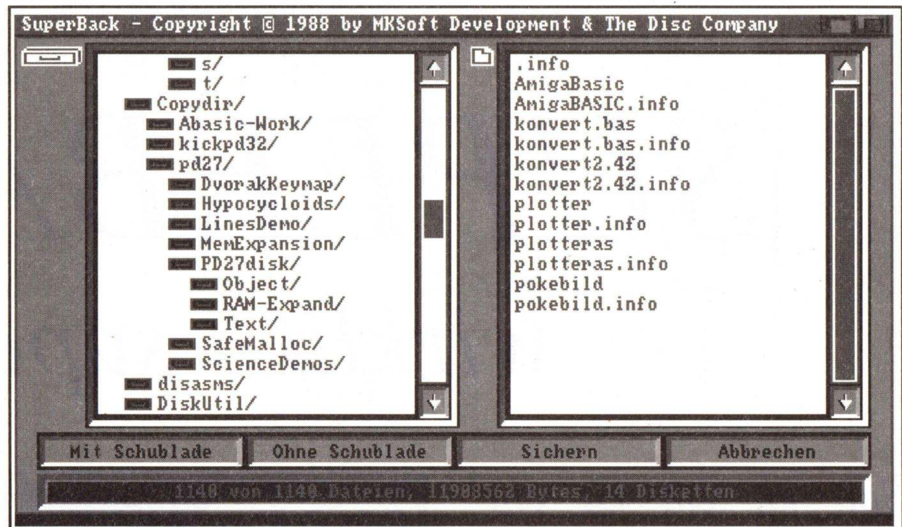


Bild 2: Der Fileselektor von Superback

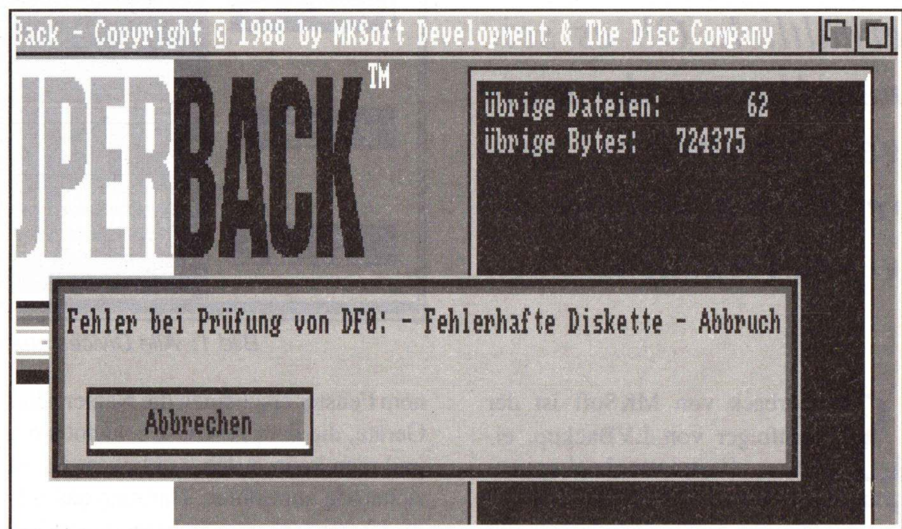


Bild 3: Onewayticket- Ein Diskettenfehler führt zum Abbruch des Backups- Ein Fortfahren ist nicht möglich

Bild 4: Man beachte die Anzahl der Bytes- bei einer 20MByte-Platte

Diskettenbedarf

Die Anzahl der benötigten Disketten lag bei obiger Platte mit Superback bei 14, mit Quarterback waren es 13 Disketten. Superback zeigt vor Beginn der Sicherung an, wieviele Disketten man benötigt, was sehr nützlich ist. So kann es nicht geschehen, daß man nach einer Stunde Backup mit einer Diskette zuwenig dasteht.

Dokumentation

In der Packung, die übrigens etwa um den Faktor 4 zu groß geraten ist (ich vermisse den Hinweis, daß der Freiraum im Karton beim Füllen technisch notwendig ist, siehe Waschmittelverpakkung), findet man neben der Diskette auch ein Handbuch, das in Englisch, Französisch und Deutsch jeweils 35 Seiten Hinweise zur Bedienung des Programms und zur Vorgehensweise beim Backuppen schlechthin gibt. Das Handbuch ist verständlich und in gutem Deutsch (keine Japanerübersetzung) geschrieben. Man findet alle Angaben, die zur Bedienung des Programmes notwendig oder hilfreich sind.

Fazit

Insgesamt kann Superback nicht an Quarterback heranreichen. In den Punkten Geschwindigkeit, Selektionsvielfalt und Disketten bedarf wird es von diesem eindeutig geschlagen. Zudem ist das Verhalten bei Diskettenfehlern nicht zu billigen. In seiner jetzigen Form wird Superback sicherlich kein Geheimtip.

SUPERBACK

+ einfach zu bedienen

- fehlerhafte Anzeigen
- Abbruch des Backups bei Diskettenfehlern
- keine Einflußmöglichkeit auf Archivebit

ENDE

VON FLORIAN DU BOIS

Fraktale Kunst

Fraktalgenerator 3D

Nachdem der Mathematiker Benoit B. Mandelbrot die nach ihm benannte Mandelbrotmenge im Zentrum der komplexen Zahlenebene fand, wurden für viele Computer Programme entwickelt, mit denen man diese Mengen untersuchen kann. Fraktalgenerator ist ein neues Programm dieser Art für den AMIGA.

Das Besondere an der Mandelbrotmenge sind die immer wieder neu entstehenden Strukturen. Wird nämlich ein Bereich vergrößert, sind immer detailliertere Strukturen von seltener Vielfalt zu entdecken. Im deutschsprachigen Raum wird diese Menge auch oft als "Apfelmännchen" bezeichnet.
Später wurden auch noch die sogenannten Juliamengen (Bild 2) entdeckt, die ähnliche Eigenschaften besitzen wie die Mandelbrotmenge. Mit Hilfe von Fraktalgenerator läßt sich der AMIGA als eine Art Mikroskop zur Untersuchung der genannten Mengen einsetzen.

Berechnung

Fraktalgenerator zeichnet sich besonders durch seine kurzen Wartezeiten aus. So dauert die Berechnung des Grundapfelmännchens zum Beispiel nur 7 Sekunden. Daß sämtliche Bilder mit 32 Farben berechnet werden, scheint dabei der Rechengeschwindigkeit nicht zu schaden. Die Möglichkeit, jede Berechnung zu jedem beliebigen Zeitpunkt abbrechen zu können, erspart oft, gerade bei "tieferen" Bereichen, bei denen die Berechnungen länger dauern, ein langes Warten.

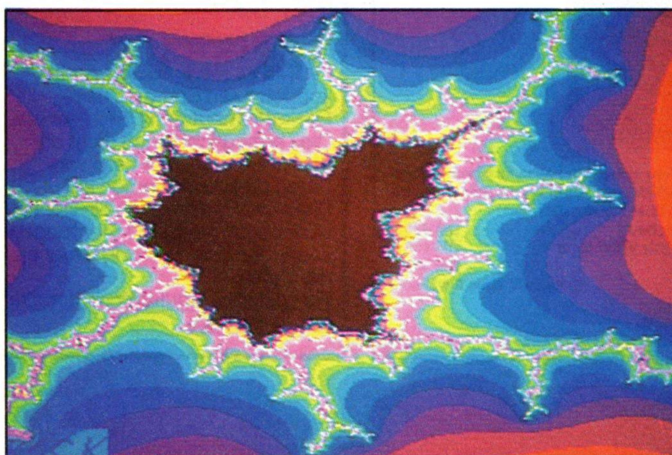


Bild 1: Auch in tieferen Bereichen einer Menge (hier Mandelbrot) kommt ein Bild mit vielen Farben zustande.

Darstellung

Nachdem entschieden wurde, ob Bilder der Mandelbrot- oder der Juliamengen berechnet werden sollen, wird automatisch erst die Grundfigur der jeweiligen Menge berechnet. Soll ein bestimmter Bereich vergrößert werden, kann dieser einfach mit Hilfe der Maus auf dem Bildschirm umrahmt werden. Bei der folgenden Berechnung füllt der ausgewählte Bereich dann den ganzen Bildschirm (Bild 4). Da die Berechnungszeit mit der Anzahl der bereits vergrößerten Bereiche zunimmt, gibt es die Möglichkeit, sich vor einer bildschirmfüllenden Berechnung ein kleines Bild berechnen zu lassen. Dies dauert nicht sehr lang und gibt einen guten Überblick über den neu ausgewählten Bereich. Es muß dadurch nicht immer gewartet werden, bis der ganze Bildschirm gefüllt ist. Hervorzuheben wäre an dieser Stelle noch, daß ein neuer Ausschnitt sogar gleich aus diesem verkleiner-

ten Bild gewählt werden kann, wodurch schnelleres Suchen nach schönen Strukturen möglich wird. Damit die Verzerrung eines Bildes ausgeschlossen wird, gibt es als Hilfe die Proportionaloption, die dem Ausschnitt immer das gleiche Verhältnis zwischen Höhe und Breite vorschreibt. Die gewünschten Koordinaten des zu berechnenden Teiles einer Menge können natürlich auch über die Tastatur eingegeben werden. Schade ist, daß es nur eine Auflösung von 320 * 256 Punkten gibt.

Die Bedienung

Fraktalgenerator ist nach kurzer Zeit sehr schnell zu bedienen, da sämtliche Optionen außer über die Maus auch über die Tastatur angewählt bzw. gestartet werden können. Die Maus wird dadurch nur noch zum Umrahmen eines Bereiches benötigt.

Die Farben

Die volle Farbenpracht eines Bildes ist auch bei tieferen Bereichen, wo die meisten Programme nur noch wenige Farben zeigen, gewährleistet (Bild 1+2), weil die maximale Rechentiefe verstellbar ist (entspricht der Zahl der maximalen Iterationen). So sind auch Strukturen zu finden, die in Bereichen liegen, wo andere Programme nur noch einen einfarbigen Bildschrim zustande bekommen würden. Passen die eingestellten Farben nicht zu einem Bild, können sie mit Hilfe des Farbeneditors verändert werden. Dieser Editor hat die üblichen Funktionen wie Spread (erzeugt einen Farbverlauf zwischen zwei Farben), Copy (kopiert eine Farbe in ein anderes Farbregister) und Range (wird beim Abschnitt Animation noch näher erklärt). Sie können auch (ohne Bild) gespeichert und geladen werden, so daß sich schöne Farbkombinationen sichern und auf andere Bilder übertragen lassen.

3D-Funktion

Bei vielen Programmen dieser Art ist eine 3D-Option vorhanden. Doch meistens sehen die damit erzeugten Graphiken nicht sehr schön aus, weil auf ihnen kaum noch etwas zu erkennen ist. Fraktalgenerator bietet hier jedoch verstellbare Parameter. Zum einen gibt es die Stufenhöhe, die den Abstand der einzelnen Terrassen angibt. Des weiteren können X- und Y-Blickwinkel eingestellt werden. Der X-Blickwinkel bestimmt, um welchen Winkel die Figur horizontal um ihren Mittelpunkt gedreht werden soll, und der Y-Blickwinkel bestimmt, um wieviel Grad die Blickrichtung vom Lot abweicht. Auch bei dieser Funktion ist die Rechenzeit beachtlich. Durch diese Einstellmöglichkeiten lassen sich auch mit der 3D-Option sehr interessante Bilder berechnen (Bild 3).

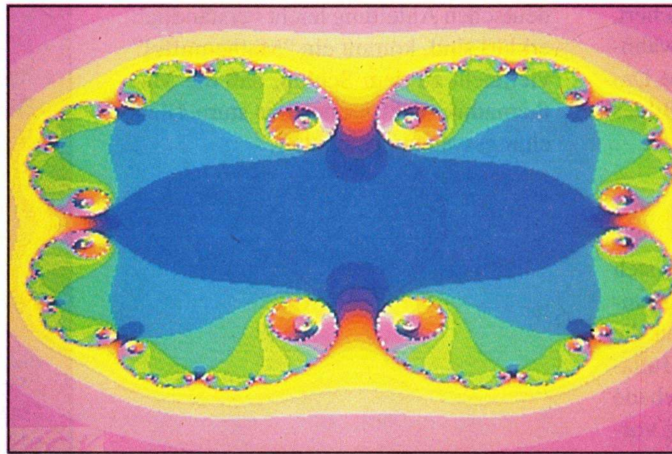


Bild 2: Auch die nicht so bekannten Juliamengen sehen sehr interessant aus.

Bild 3: Die dreidimensionalen Graphiken können durch geeig - neteWahl der Parameter sehr interessant aussehen.

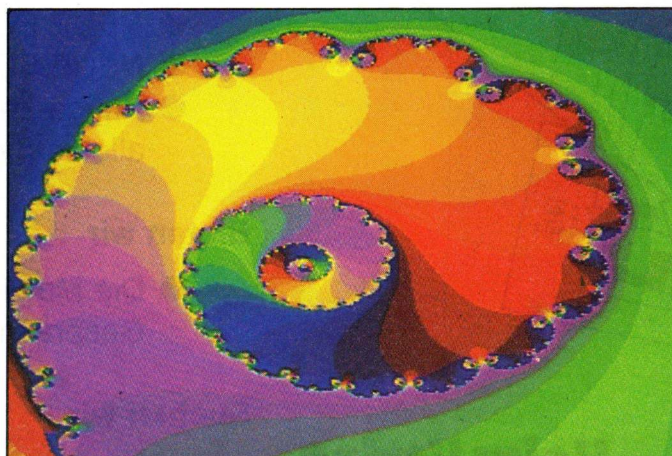


Bild 4: Eine vergrößerte Spirale aus Bild 2 ohne Farbenverlust.

Animation

An Animationsmöglichkeiten stellt das Programm zwei zur Verfügung. Die eine Möglichkeit wird auch bei Malprogrammen wie zum Beispiel bei DPAINT verwendet. Hierbei werden einfach die Farben gecycled. Mit Range im Farbeditor werden die beiden Randfarben festgelegt, zwischen denen die anderen cyclen sollen. Die zweite Möglichkeit ist, Bilder, die sich in verschiedenen Bildpuffern befinden, nacheinander auf dem Bildschirm erscheinen zu lassen. Bildpuffer sind Bereiche im Speicher, in denen sämtliche Daten eines bestimmten Bildes abgelegt werden können. Es wird also nicht nur ein erzeugtes Bild abgelegt, sondern mit ihm auch seine spezifischen Daten (Koordinaten, Blickwinkel u.s.w.). Zwischen den einzelnen Bildpuffern kann auch sonst manuell hin- und hergeschaltet werden, wodurch eine gleichzeitige Bearbeitung mehrerer Bilder möglich wird.

Laden und Speichern

Berechnete Bilder lassen sich im IFF-Format abspeichern, so daß deren Weiterverarbeitung durch Graphikprogramme möglich wird. Es werden aber nicht nur die Bilder als solche abgespeichert, sondern auch (wie bei den oben erwähnten Bildpuffern) deren spezifische Daten, so daß nach dem Laden eines Bildes gleich weitergeforscht werden kann.

Fazit

Bei Fraktalgenerator 3D handelt es sich sicherlich um eines der besten Programme für den AMIGA, mit dem man die fraktale Welt der Mandelbrot- und Juliamengen erforschen kann. Sehr schade finde ich, daß nur eine Auflösung vorhanden ist, denn manchmal stören die "Treppen" in den Bildern schon ganz schön. Vielleicht wird dieser Punkt in einer möglichen späteren Version noch geändert. Die Schnelligkeit sowie die Genauigkeit der einzelnen Berechnungen sind jedoch erstaunlich. Da die einzelnen Optionen in der beiliegenden deutschen Anleitung leicht verständlich erklärt sind, kommt ein "Mathemuffel" vielleicht zu dem Schluß, daß die Mathemathik nicht immer nur sture Gleichungssysteme beinhaltet.

3D FRAKTAL-GENERATOR

+ sehr schnell
+ Pal-Auflösung
+ Iterationstiefe verstellbar
+ Bilder im IFF-Format
+ Berechnungen immer abbrechbar
+ berechnet auch Juliamengen
+ deutsche Anleitung

- nur eine Auflösung

Bezugsadresse:
DMV-Verlag,
Postfach 250,
3440 Eschwege
Preis: DM 69.-

ENDE

VON LARS SCHMITT

MODEM TALKING

Mailboxen im Test

Und weiter geht's, in diesem Monat wollen wir Ihnen weiter Mailboxen vorstellen. Wir beginnen im hohen Norden

VMSR

Die VMSR ist ein befehlsorientiertes System. Durch die relativ geringe Anzahl an Befehlen kann man sich aber in ihr recht schnell zurechtfinden. Wenn ich allerdings so nachdenke, was ich in meinem ca. halbstündigen Test so alles an aktuellen Info's gelesen habe, fällt mir ehrlich gesagt nichts mehr ein. Es ist, so meine ich, schon ärgerlich, wenn der Gebührenzähler qualmt, während man sich 'Softwareprobleme' vom MÄRZ 1988 durchliest (Der Testtermin war der 01.01.1989.) ..! Obwohl die Anruferzahlen nicht gerade als niedrig zu bezeichnen sind und die Box schon seit über 4 Jahren Online ist (für Mailboxen ein sehr hohes Alter!), scheint sich das Publikum auf reines Konsumieren zu beschränken. Aus der nebenstehend abgedruckten Liste können Sie selbst entnehmen, daß die Mailbox-Boards im Schnitt 2 Monate ohne Update vorsichhingammeln! Leider ist die Übertragungsrate der VMSR auch nicht 300Baud, wenn's um Diskettenzugriffe geht! Auch sonst kann das Mailboxsystem mit keinen nennenswerten Besonderheiten aufwarten.

Interessant wird die VMSR wohl erst wieder in einem halben Jahr, denn dann soll ein völlig neues Programm (auf einem XT) ans Netz gehen. Und was ich darüber so alles gehört habe, hört sich wirklich gut an (also das Nümmerchen sollte man sich einmal irgendwo merken).

In der jetzigen Version ist die VMSR allerdings nur negativ zu beurteilen. Da kann auch die Idee eines 'Software-Problem-Boards', die nicht gerade alltäglich ist, nicht viel ändern, wenn der letzte Eintrag - wie gesagt - NEUN Monate alt ist!!

Fazit: Bis zur Revision im Sommer kann man die VMSR getrost von der Liste der Stammailboxen streichen!

VMSR Mailbox-System

Steckbrief:

Mailbox:	040-816132 24h 300bd fd 7/N/1
Programm:	BASIC compiliert (mit Maschinenspracheroutinen)
Hardware:	C=128 (im 64'er-Modus), 2 Floppies 1541, EPSON-Koppler
	- langsame Ausgabe - veraltete Einträge - kleine Auswahl - veraltetes Konzept!
GESAMT:	mangelhaft (im Sommer wieder anrufen)

```
                    D i e

VVV    VVV  MM M MM    SSSSSSSS RRRRRRRR
VVV    VVV MMMMMMMMM SSSSSSSSS RRRRRRRRR
VVV    VVV MMMMMMMMM SS        RRR    RRR
VVV    VVV MMM M MMM SSSSSSSS  RRRRRRRRR
VVV    VVV MMM M MMM  SSSSSSSS RRRRRRRR
 VVV VVV   MMM   MMM        SS RRR RRR
  VVVVV    MMM   MMM SSSSSSSSS RRR  RRR
   VVV     MMM   MMM SSSSSSSS  RRR   RRR

              U s e r - B o x

                 ist Online
```

```
Steuerzeichen:     CTRL X/S/Q/C/N

Die VMSR-USER-BOX Basiert auf der
Version 5.7 VON  [THW]  of  M.C.S.
umgeschrieben und  ueberarbeitet von
[RCV] SYSOP der VMS-USER-BOX V6.xx
ueberarbeitet von [VMSR] SYSOP der
VMSR-USER-BOX V6.13


Du bist der  24796. Anrufer bei V M S R

Sonntag den 01.01.1989 / 20.55.21 Uhr


 Menues:
 -------
```

```
HELP          Anleitung der Box
?             Menue-Helplisten
IN            Info-Menue
MA            Mailbox-Menue
ML            Message-Ausgabe-Menue
ME            Message-Eingabe-Menue
SPR           Software Probleme & Loes.
HINFO         Hacker-Info
              (Passwortabfrage)
GF            Gemeinschaftsfaecher
              (NICHT ALLGEMEIN!!!)

   HAUPT-MENUE
   -----------

TI            Time/Datum/Online
UID name,pw   User-ID eingeben
oder
UID name      Username (PW ohne Echo)
NU            NEUER USER!!!!!!!!!!!
PRO           Profi on/off
SOR           System Operater rufen
DIR           Message Directory
BE            Bell on/off
LF            Linefeed on/off
MSO           Message an den SysOp
LOGOFF        Verbindung abbrechen

   INFO-MENUE
   ----------

MAR           Markt (gewerblich)
CL            VMS-User-Club
AK            Aktuelles
ALG           Allgemeines
TEL           Telefonliste von Mailboxen
NUA           NUA-Liste
RI            Rechner-Info

   MAILBOX-MENUE
   -------------

PML           Persoenlich Message lesen
PME .....     Persoenlich Message eingabe
UL            Userliste gesammt
ULN           Nur Namen anzeigen
ULN .....*    Namen suchen mit Joker
ULN .....     Nach Namen suchen
ULS .....     Nach Stadt suchen
ULR .....     Nach Rechner suchen
ULD .....     Nach Datum suchen
ULT .....     Nach Status suchen
NU            Neuer Usereintrag
UIDA          User Daten aendern
MSO           Message an den SysOp

  MESSAGE-AUSGABE-MENUE
   ---------------------

DIR           Message / Info Directory
SU            Suche
BI            Biete
KO            Kontakte
AL            Allgemeines
JC            Jokes Corner
BR            MSG vom SysOp

   MESSAGE-EINGABE-MENUE
   ---------------------

SU            Suche
BI            Biete
KO            Kontakte
AL            Allgemeines
JC            Jokes Corner

   Software Probleme & Loesungen
   -----------------------------

SPL           Software Probleme lesen
DIS           Diskusionsbeitraege lesen
ESPL          Software Probleme ein-
              geben
EDIS          Diskusionsbeitraege ein-
              geben

   Gemeinschafts Faecher
   ---------------------

(Diese Faecher sind nicht allgemein
zugaenglich!!!!!)

LFA             Lesen Fach
EFA             Eingabe Fach
SFA             Fachinhalt loeschen
FDIR            Faecherdirectory

MSG-Texte  / Directory

Suche      :28.10.1988 / 14.49
Biete      :17.10.1988 / 20.06
Kontakte   :18.12.1988 / 20.14
Jokes      :26.10.1988 / 21.08
Allgemeines:13.12.1988 / 11.20
CL         :??.??.???? /??.??
MSG v.SYSOP:18.10.1987 / 20.03
User       : 193
Anrufer    : 24754
```

Terra Com

Leider ist uns im letzten Artikel ein kleiner Fehler unterlaufen. Die Telefonnummer im Steckbrief zum Terra-Com Mailbox-System wurde falsch abgedruckt. Findige DFÜ'ler haben die korrekte Nummer sicherlich schon im Probelisting der Mailbox erspäht. Hier ist sie noch einmal:

(06142-5615289)

Wir wünschen eine gute Verbindung.

LOGOFF

TeCoS

Als einer der ältesten privaten Computerclubs wurde der PTC vor ca. 10 Jahren gegründet. Aus der Handvoll Mitglieder hat sich ein Club mit einer Vielzahl von Mitgliedern (von Freaks bis zu normalen Anwendern) entwickelt. Eine Leistung des Clubs ist die Clubzeitschrift INFO und die zugehörige INFO-DISK. Im Jahre 1984 wurde die Mailbox TeCoS (Tele Communications System) ins Leben gerufen. Hauptziel dieser Mailbox war es, Mitgliedern das Knüpfen von Kontakten zu erleichtern, und die Programme der INFODISK über die DFÜ abrufbar zu machen. Auch der TeCoS wuchs rasch und bietet heute mit 16 MBytes an Text- und Programm-Files fast alle erdenklichen Informationen zu allen Rechnern an! Beim ersten Besuch der TeCoS sollten Sie alle Texte und Menüs abspeichern. TeCoS ist befehlsorientiert aufgebaut und darum für Neulinge etwas schwerer erlernbar. Hat man sich allerdings an die Syntax der Befehle gewöhnt, bietet ein solches befehlsorientiertes System seine ganz klaren Vorteile! Eine Eintragung als Benutzer ist kostenlos. Loggen Sie sich einfach einmal in den TeCoS ein (nicht mit GAST oder so, sondern mit einem richtigen Namen/Pseudonym). Senden Sie dann eine Postkarte an den PTC, Kantstraße 12, 6050 Offenbach, auf der sinngemäß steht: Bitte tragt mich ein. Mein Name ..... Meine gesamte Anschrift ..... meine Telefonnummer ..... Mein Paßwort ..... Den Rest macht der SysOp so ca. in einer Woche!

Anzumerken sei noch:

- Loggen Sie sich bei TeCoS nie mit GAST / HACKER / FUZZI oder ähnlichem Mist ein, das mag er nicht!
- Sichern Sie Ihre Kommunikation zum späteren Ausdrucken!
- Sehen Sie sich vor dem ersten Besuch erst einmal die hier abgedruckten Menüs und Flußdiagramme an. In ihnen sind nahezu alle Befehle des TeCoS aufgeführt und kurz erläutert. Durch die Eingabe von?<Befehl> kann man sich einen Befehl auch erläutern lassen. Z.B. erklärt ?L den Befehl LESEN im Hauptmenü!

(Lars Schmitt)

```
 HAUPTMENUE  (* = nicht fuer Gaeste)

  >>> BRIEFE <<<                                >>> UNTERMENUEs <<<
* E)ingabe    eines Briefes *                   T)exte   (Referate) lesen
* S)treichen  Loeschen alter Briefe             F)ile -  Transfer
  P)ersoenl.  Brief da? (von mir/an mich)       U)tility Parameter anzeigen/aend.
  L)esen      von Briefen
  I)nhalt     (Absender/Zeit/Betreff)           >>> KONFERENZEN <<<
  B)etreffs   listen (QuickScann)             * K)onferenz-Modus aufrufen

  >>> SONSTIGES <<<                             >>> VERABSCHIEDUNG <<<
  N)achricht  an den SysOp schreiben            G) Goodbye (EXIT, LOGOFF usw.)
  A)ktuelles  Last News (nochmals) lesen        ==============================
  H)ilfe      zum Hauptmenue
  X)perten    -Modus ein/aus
  Z)eilen pro Bildschirmseite (0 = endlos, 22 = Default, 0..255 moeglich)
 ?) Hilfe zu den einzelnen Moeglichkeiten des Hauptmenues
 ---------------------------------------------------------------------------


Hauptmenue <A,B,E,F,G,H,I,K,L,N,P,R,S,T,U,X,Z,?(Bu)>=> t

Damit es nicht zu unuebersichtlich wird, haben wir die Uebersicht
(Inhaltsangabe) etwas aufgegliedert.
                                                                               *
Text  Uebersicht  Mindestlevel fuer       *   FUEHRENDE NULLEN MUESSEN
Nr.   ueb. Texte  Aufruf dieser Texte:    *   ========================
-------------------------------------     *   N I C H T  MITGETIPPT
 1    0000 - 0999  05  (Gast)             *
 2    1000 - 1999  10  (eingetr. User)    *   WERDEN: STATT  "0001"
 3    2000 - 2999  30  (PTC-Mitgl.)       *
-------------------------------------     *   NUR EINE "1" TIPPEN!

Bitte waehlen Sie zwischen 1 und 3 fuer eines der Inhaltsverzeich-
nisse oder, Ihrem Level entsprechend, direkt die NUMMER des
gewuenschten Textes, OHNE 'T' oder 'TEXT'vorweg!



FILE-TRANSFER:     (* = nicht fuer Gaeste)

 SEHEN:                                         LADEN/SPEICHERN
 -------                                        ---------------
 L)iste der moegl. Directories                * D)ownload (TECOS -> SIE)
   bzw. L;xx Auflisten DIRxx                  * U)pload   (Sie -> TECOS)
 N)eue Files anzeigen seit TTMMJJ
 S)uche nach File-Namen (Schlagwort)           SONSTIGES:
                                               ----------
 HILFEN                                         X) Epertenmodus ein-/
ausschalten
 ------                                         Z)urueck zum Hauptmenue
 H)ilfe kurze Erlaeuterung der Befehle          und
?) ausfuehrliche Hilfe zu den einzelnen        G)oodbye (unser LOGOFF, EXIT,
    Moeglichkeiten des Filetransfer              LOGOUT, ENDE usw.
 ---------------------------------------------------------------------------


 UTILITIES:    (* = nicht fuer Gaeste erreichbar)

  >>> AENDERN <<<                              >>> ANZEIGEN <<<

  F)ile Uebertragungs-Parameter                *U)nser LOGBUCH (Anrufer od. User)
  G)raphik-Modus                               *S)ystem-Statistik (mini)
  L)ine Feeds (ein/ausschalten)                D)auer der Verbindung
  N)ULLS = Warten nach CR                      E)instellung (IHRE PARAMETER)
  R)echter Rand bei EINGABE setzen
* P)assword aendern                            >>> BESONDERES <<<
  !) Prompter-Klingel ein/ausschalten
  X) Experten-Modus ein/ausschalten            H) Hilfe
  Z) Zurueck zum Hauptmenue
  ?) ausfuehrliche Hilfe zu den einzelnen Punkten dieses Menues
 ---------------------------------------------------------------------------


INLOG- UND AUSWAHLPROZEDUR TECOS (Flowchart)

                    ---------------------
                    - ERKENNEN  CARRIER -
                    -PARAMETER-ERKENNUNG-
                    -    (automatisch)  -
                    ---------------------
                              -
                     ---------------
                     - STARTGESULZE -
                     ---------------
```

TeCoS Mailbox-System

Steckbrief:

Mailbox :	069/8003539/24h/ 300bdfd 7/N/1
Programm:	Compiliertes BASIC (ca. 300KByte) mit ASSEMBLER-ROUTINE
Hardware:	IBM/XT mit zwei 20M Byte-Platten, 720K und 360K Floppies
Betreiber:	PTC Kantstraße 12 6050 Offenbach
POSITIV :	Unmengen von Info /geduldiger SysOp/schnelle Ausgabe/viele Befehle (Suchbefehle ect.)
NEGATIV :	Durch die vielen Befehle für Neulinge schwierig
GESAMT :	Sehr empfehlenswert!

LOGOFF

```
                 ---------------------------
                 -      ANFORDERUNG von    -
                 - Vorname/Name(Pseudonym) -
                 ---------------------------
                              -
                              -
                       --------------
              ---------erster Anruf?---------
              -        --------------       -
              - Ja                   Nein   -
  -------------------------    -----------------------
  -      EINTRAGEN        -    -      ANMELDEN       -
  -------------------------    -----------------------
  -Ihr Passwort auswaehlen -    - Ihr Passwort als    -
  -     und eingeben       -    - Zugangsberechtigung -
  -                        -    -      eintippen      -
  - Fragen zu Ihrem System -    -----------------------
  - beantworten (einmalig) -                -
  -------------------------                 -
              -                             -
              -------------------------------
                              -
                              -
                 ---------------------------
                 - Letzte Neuigkeiten lesen -
                 -        (LAST NEWS)       -
                 ---------------------------
                              -
                              -
            ---------------------------------------
            - Anzeige der Messages, die von Ihnen -
            -  kommen bzw. fuer Sie bestimmt sind -
            -   (wenn solche MSG vorhanden sind)  -
            ---------------------------------------
                              -
                    ---------------------
                    - H A U P T M E N U E -
                    ---------------------
            ------------  -  -  -  ------------
            -              (T) - (U)          -
------------------------- - - -  ------------------------
-    NACHRICHTEN-TEIL    - - (F) -  -       sonstige        -
-    (BRIEFE)            - - - -    - Auswahlmoeglichkeiten-
------------------------- - - -    ------------------------
-  E  Eingabe  (+)       - - - -    - N Nachricht an den   -
-  S  Loeschen  (+)      - - - -    -   SysOp schreiben    -
-  I  Kurz-Inhalt        - - - -    - Z Zeilenzahl waehlen -
-  B  Betreffs listen    - - - -    - A Last News wiederh. -
-  L  Lesen von MSG      - - - -    - H Hilfe zu d. Bef.   -
-  P  pers. Mail listen  - - - -    - X Expertenmodus ein- -
------------------------- - - -    -   od. ausschalten    -
                          - - -    -?<Bu> Hilfe zu einen -
     -----------------   - - -    -       Befehl          -
-------------------         - -    - G Goodbye/LOGOFF    -
-   TEXT- MENUE   -         - -    -  Ende der Verbindung-
- grosse Texte lesen-       - -    ------------------------
-------------------         - -
- 0  Inhaltsver-  -         -  ----------------
- :  zeichnisse der -       -                -
- 3  einzelnen Files-       -    ------------------------
-- - - - - - - - --         -    -UTILITIES = Hilfs- u.-
-10               -         -    -    Aenderungsmenue   -
- :  div. Text-Files-       -    ------------------------
-3999             -         -    - Ae N D E R N   V O N -
-------------------         -    -- - - - - - - - - - --
                            -    - F File-Uebertra-     -
                            -    -   gungsparametern    -
                            -    - N NULLS senden       -
         ------------------      - R rechter Rand bei   -
-----------------------------    -   Ihrer Eingabe      -
- F I L E - T R A N S F E R -    - P PASSWORT           -
- unlimitierte Datenuebertr.-    - X Expertenmodus      -
- von Programmen und Dateien-    -! Eingabe-Bimmel      -
-----------------------------    - G Grafik aendern     -
- L Liste der (virtuellen)  -    - L Line-Feed ein/aus  -
-   Directories im TECOS    -    -- - - - - - - - - - --
- N Neue Files anzeigen seit-    - A N S E H E N   V O N-
-   Datum MMTTJJ (am. Norm) -    -- - - - - - - - - - --
- S Suche nach Filenamen od.-    - E Report Ihrer derz.-
-   Teilen davon (Lexikon)  -    -   eingest. Parameter-
- D DOWNLOAD (TECOS -> Sie) -    - U TECOS-LOGBUCH (An--
- U UPLOAD (Sie -> TECOS)   -    -   (rufer od. USER    -
- X Expertenmodus schalten  -    - S Systemstatistik    -
- Z zurueck zum Hauptmenue  -    - H Hilfstexten        -
- ?<Bu> Hilfen zu Topic     -    -- - - - - - - - - - --
- H Hilfe zu den Befehlen   -    -  S O N S T I G E S   -
- G Godbye/LOGOFF/LOGOUT    -    -- - - - - - - - - - --
-   Verbindung mit TECOS    -    - Z zurueck zum Haupt--
-   beenden                 -    -   menue (QUITT)      -
-----------------------------    ------------------------
```

VON ANDREAS KRÄMER

Englisch für Anfänger

Die englische Sprache ist neben der spanischen die verbreitetste auf der Welt. Nicht nur aus diesem Grund wird sie in den Schulen gelehrt. Doch können manche Schüler mit Englisch nicht viel anfangen und tun sich schwer, was sich natürlich in der Zeugnisnote bemerkbar macht. Um mit seinem Hobby, dem Computer, die Englischkenntnisse aufzubessern, dient das Programm ENGLISCH KURS 1.

Wer einen typischen Vokabeltrainer erwartet, der liegt gänzlich falsch. Mit dem ENGLISCH KURS 1 aus dem Hause CDC präsentiert sich eine eigenwillige Variante von Lernsoftware. Der Kurs wartet mit viel Grafik und Sound auf, dem Lernwilligen soll die Sprache auf spielerische Art und Weise beigebracht werden. Viele Bilder, im Comicstil gezeichnet, begleiten das Programm. Die Bedienung geschieht ausschließlich mit der Maus.

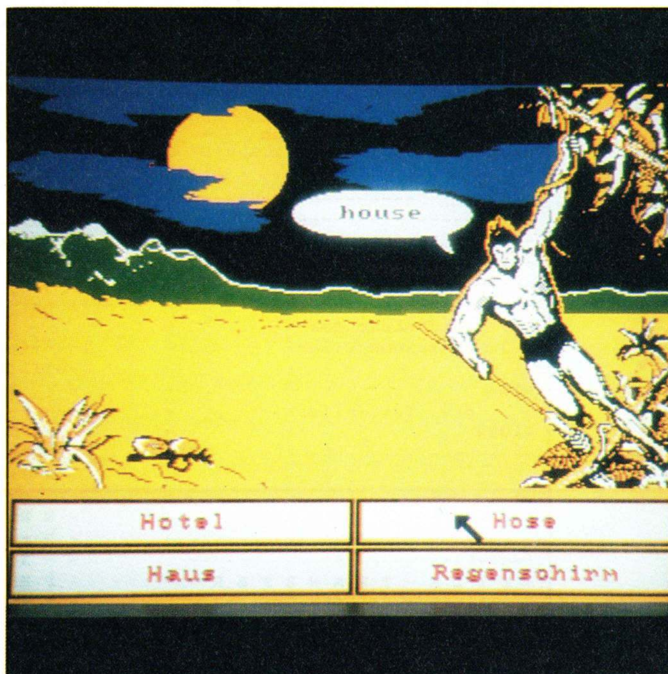
Neben einer Diskette erhält der Kunde eine magere Anleitung, die lediglich die Grundzüge des Programms erläutert. Das Programm kann nur durch Einlegen der Diskette nach einem Reset oder Einschalten des Rechners gestartet werden, ein Starten von der Workbench ist nicht möglich. Nach einer langen Ladezeit erscheint endlich das Hauptmenü, vier Menüpunkte stehen zur Auswahl.

1. Kurs

Hinter diesem Menüpunkt verbirgt sich ein in 12 Lektionen unterteilter Lehrgang. Es werden Grammatik, Satzbau, Redewendungen, usw. erklärt. Die Schwierigkeit nimmt von Lektion zu Lektion stetig zu. Die erste Lektion beinhaltet beispielsweise nur eine kleine Einführung, viel Lernen kann der Schüler hier noch nicht, auch in der zweiten und dritten Lektion bekommt man nur Grundsätzliches über die englische Sprache mitgeteilt. In der Lektion 12 werden dann schon komplexere grammatikalische Grundzüge erklärt. Jede Lektion ist mit einem Comicbild unterlegt. Das hat leider manchmal den Nachteil, daß der Text schlecht zu lesen ist.

2. Spiel

Hier kann der Schüler das erlernte auf spielerische Art überprüfen. Werden jedoch zuviele Fehler begangen, kommt es zu einem geschmacklosen Abbruch des Spiels (Das hätte der Programmierer lieber weglassen sollen!). 12 Stages mit je 15 Fragen beinhaltet das Spiel. Der Schwierigkeitgrad wächst mit zunehmender Spieldauer. Vier Antworten stehen zur Auswahl, wobei nur eine richtig ist. Auch hier sind Bilder untergelegt, die bei jedem Stage gewechselt werden.



3. Vokabelliste

Der Anwender kann sich alle Vokabeln und einige Redewendungen des Kurses chronologisch ausgeben lassen.

4. Anleitung

Durch Anwählen dieses Punktes wird eine kurze Erklärung des Programms ausgegeben.

Fazit

Der ENGLISCH KURS bietet eine interessante Alternative zu den konventionellen Vokabeltrainern. Die extravagante Aufmachung und Bedienung des Programms sind nicht jedermanns Sache, auch ist der Kurs lediglich für Personen gedacht, die keinerlei oder sehr wenig Englischkenntnisse besitzen. Für jüngere Schüler, die ab der 5. Klasse das erste Mal Englich als Unterrichtsfach bekommen, ist der Kurs aber eine gute Unterstützung. Den jungen Schülern/innen wird die Aufmachung eher zusagen als älteren Semestern. In Vorbereitung ist der ENGLISCH Kurs 2. Eltern, die ihren Kindern ein wenig unter die Arme greifen wollen, können das Programm durchaus kaufen. Der didaktische Aufbau des Kurses ist durchdacht und ausgeklügelt, sofern der Schüler den Kurs konzentriert abarbeitet. Daß keinerlei Manipulationsmöglichkeiten des Kurses gegeben sind, muß als ein kleines Manko gesehen werden. Ist der Kurs abgearbeitet, wandert das Programm unweigerlich in die Ecke und wird nur von einem neuen Lernwilligen wieder hervorgekramt.

Vertrieb:
CDC
Louisenstr. 115
6380 Bad Homburg
Tel. 06172-24728
Preis 49.90 DM

AZTEC-C FÜR AMIGA VERSION 3.6

Wußten Sie, daß eines der verbreitesten und komplexesten Betriebssysteme - UNIX - in C geschrieben ist ?

Wußten Sie, daß auch das Betriebssystem des AMIGA größtenteils in C geschrieben wurde ?

Wußten Sie, daß C eine der wichtigsten und modernsten Programmiersprachen ist ?

Wußten Sie, daß Aztec-C einer der schnellsten und leistungsfähigsten Compiler für den AMIGA ist ?

Wußten Sie, daß jetzt Aztec-C in der Version 3.6 verfügbar ist ?

Möchten Sie mehr darüber wissen? Dann schicken Sie uns einen ausreichend frankierten Rückumschlag und Sie erhalten ausführliche Information.

HANNOVER MESSE
CeBIT '89
Welt-Centrum Büro · Information · Telekommunikation
8. – 15. MÄRZ 1989
HALLE 8 / E42

HIERMIT BESTELLE ICH:

- ☐ AZTEC-C68K/AM-P PROFESSIONAL SYSTEM FÜR DM 298.-
- ☐ AZTEC-C68K/AM-D DEVELOPER SYSTEM FÜR DM 598.-
- ☐ AZTEC-SDB SOURCE LEVEL DEBUGGER FÜR DM 149.-

Versandkosten: Inland DM 7,50 Ausland DM 10,-
Auslandbestellungen nur gegen Vorauskasse
Nachnahmegebühr DM 3,70

- ☐ Vorauskasse
- ☐ Nachnahme

NAME: ____________

VORNAME: ____________

STRAßE: ____________

ORT: ____________

UNTERSCHRIFT: ____________

AZTEC-C IST EIN EINGETRAGENES WARENZEICHEN VON MANX SOFTWARE SYSTEMS

MAXON COMPUTER GMBH
INDUSTRIESTRAßE 26
6236 ESCHBORN
TEL. 06196/481811

PANTHERBOX

Das Panther-Box Mailbox-System gehört mit seinem dreijährigen Bestehen sicherlich zu den etablierten Mailbox-Systemen im Frankfurter Raum. Das Angebot zu allen Rechnern ist sehr umfassend (Referate in Hülle und Fülle), die befehlsorientierte Benutzeroberfläche leicht erlernbar. Trotzdem sollten Sie bei Ihrem ersten Besuch alle Ein- und Ausgaben protokollieren, um sie sich später ausdrucken zu können. Eine Befehlsübersicht ist nebenstehend abgedruckt.

Besonders interessant ist die INSTALL-Funktion. Haben Sie sich in das PMS eintragen lassen, können Sie das System auf Ihrem Terminal einrichten. Durch einfache Fragen (siehe Ausdruck) prüft das PMS, ob Sie ein Terminal mit Umlautdarstellung, Bellfunktion etc. besitzen, und speichert die so erfaßten Daten unter Ihrem Account ab. Beim Login werden diese Daten dann automatisch wieder aktiviert!

Im Ausdruck finden Sie weiterhin die Begrüßungsmeldung, die Befehlsübersicht, alle abrufbaren Pinboards und eine Referate-Übersicht.

(Lars Schmitt)

LOGOFF

Panther-Box Mailbox-System

Steckbrief:

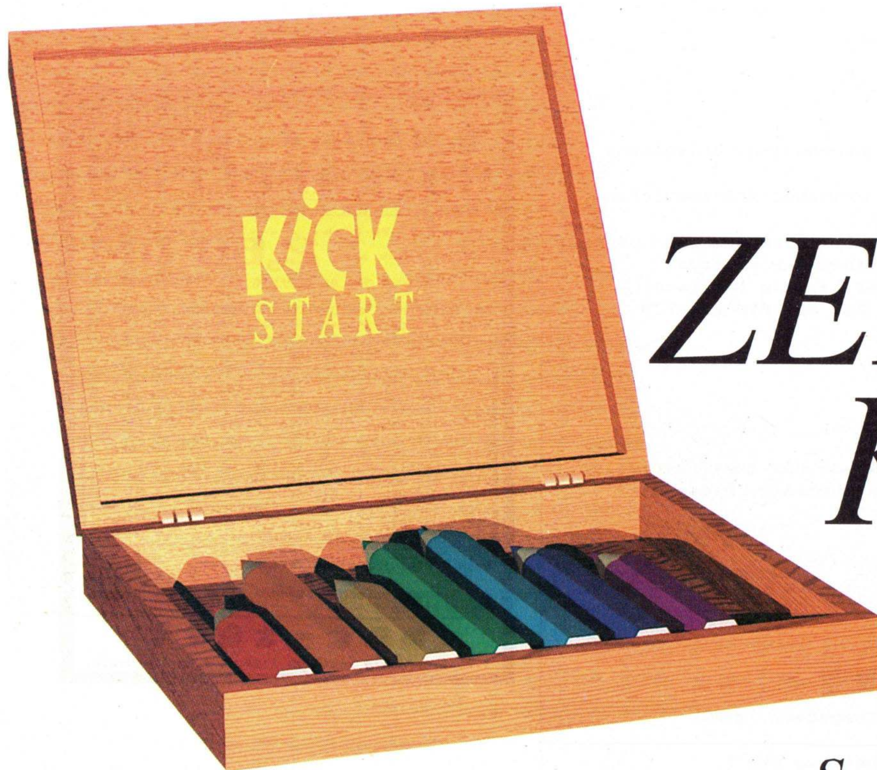
Mailbox :	06102-17328 24h 300bd fd 8/N/1
Programm:	1500-zeiliges Turbo-Pascal- Programm unter CP/M
Hardware:	Eigenbau auf Z80-Basis / 8 MHz mit 23 MByte-Platte
POSITIV :	Infos über Infos, Install-Routine, selbsterklärende Befehle (Siehe PIN- oder REF-Befehl im Ausdruck)
NEGATIV :	Durch die Befehle für Neulinge schwerer erlern bar
GESAMT :	Empfehlenswert!

```
*******************
*                 *
*  Willkommen im  *
*                 *
*  PANTHER - BOX  *
*                 *
*  Mailbox System *
*                 *
*******************

Parameter: 8 data, no parity, 1 stopbit.

HELP      <=== *** Hilfe-Funktion ***
HELP BEF  <=== Befehlsliste ausgeben
LOGOFF    <=== Beenden der Verbindung


Befehl >help bef
Die BEFEHLSLISTE des PANTHER Mailbox Systems:

Im Folgenden finden Sie eine zusammenfassende Liste aller verfuegbaren
Befehle und deren Kurzbeschreibungen. Die Teile des Befehls, die in
Grossschrift angegeben sind, muessen UNBEDINGT eingegeben werden, die
kleingeschriebenen Teile werden nicht beachtet und koennen daher wegge-
lassen werden.


Befehl                  Kurzbeschreibung
=======================================
PINboard                Gibt eine Liste aller derzeit verfuegbaren
                        Pinboards aus
PINboard <Name>         Gibt das Pinboard "<Name>" aus

SOFtbox                 Gibt eine Liste aller derzeit verfuegbaren
                        Programme aus
SOFtbox <Name>          Ruft das Programm "<Name>" ab

REFerate                Gibt eine Liste aller derzeit verfuegbaren
                        Referate aus (Siehe "HELp REFerate")
REFerat <Name>          Gibt das Referat "<Name>" aus.

MAIl                    Gibt alle empfangenen Mails aus

SENd                    Gibt Liste aller moeglichen Empfaenger aus
SENd <Parameter>        Sendet einen Brief an den SysOp, ein Pin
                        board, einen Benutzer, die Softbox oder die
                        Referatecke.

PASswort <Alt> <Neu>    Aendert das Passwort von "<Alt>" auf"<Neu>"

NEWs                    Gibt die System-News aus

GASt                    Hilft Gaesten,sich als Benutzer einzutragen

OPErator                Ruft den System-Operator (Siehe "HELp
                                                    OPErator")
USEr                    Gibt die Liste aller eingetragenen Benutzer

USEr <Stichwort>        Gibt eine Liste der Benutzer auf, bei denen
                        <Stichwort> in ID, Name, Ort oder Last Logon
                        Date vorkommt. Siehe HELP USER!

TIMe                    Gibt aus, wie lange Sie noch im System bleiben
                        duerfen

VERsion                 Gibt die derzeitige Programmversion aus.

LOGoff                  Unterbricht Verbindung (Siehe "HELp LOGoff")

INStall                 Ruft das INSTALL-Programm zur Anpassung des
                        PANTHER-Mailboxsystems an Ihr Terminal
                        (-programm) auf. Siehe hierzu "HELp INStall".

HELp                    Gibt Uebersicht ueber alle verfuegbaren Hilfen.
HELp ME                 Gibt Anleitung zum HELP-Komando
HELp *                  Gibt ALLE Hilfstexte aus.

HELp <STICHWORT>        Gibt alle Informationen ueber alle Stich-
                        woerter aus, die mit <STICHWORT> beginnen.

Die Befehle PASswort und INStall, sowie das Abrufen von Programmen aus
der Softbox, koennen nur von eingetragenen Benutzern, jedoch NICHT von
Gaesten aufgerufen werden.
```

```
Befehl >pin
Folgende Pinboards sind derzeit verfuegbar:

PINboard ADVentures        Adventure-Freak's Corner
PINboard ALLgemein         Allgemeines
PINboard SUChe             Gesuche
PINboard BIEte             Biete
PINboard HACker            Hacker's Corner
PINboard KINo              Kinoprogramme, Filmbesprechungen, Videofilme
PINboard MUSik             Alles zum Thema Musik
PINboard TEChnik           Technische und andere Computer-Informationen
PINboard VERanstaltungen   Veranstaltungskalender
PINboard ZOFf              Wie der Name schon sagt... (Diskussionen etc.)
GLObal                     Ruft alle oben aufgefuehrten Pinboards
                           nacheinander ab. GLObal ist die Kurzform
                           fuer die Befehlsfolge PIN ADV PIN ALL PIN
                           SUC etc..

Gaeste koennen nur das PIN ALLgemein abrufen.

Befehl >ref

Aus Platzgruenden und zur besseren Uebersicht wurde das Inhaltsverzeich-
nis der Referate in mehrere Einzelverzeichnisse zerteilt. Diese sind
durch die folgenden Befehlen abrufbar:

Befehl           Einzelverzeichnis
=================================

REF 001          Referate / Bauanleitungen
REF 002          Des Hacker's Lektuere
REF 003          Texte zur Weiterbildung / Sachtexte
REF 004          Rund um DATEX
REF 005          Telefonnummern von Mailboxen, Computern, etc.
REF 006          Unterhaltung
REF PMS ******* System-Informationen (Unkostenbeitrag etc.)

Wenn Sie uns einen Text zukommen lassen moechten, der hier veroeffent-
licht werden soll, geben Sie bitte "SEND REFERATE" ein.

Denken Sie bitte daran, bei Uploads von vorbereiteten Texten im Editor
das Echo abzuschalten. Naeheres hierzu durch HELP ECHO, wenn Sie im
Editor dieser Mailbox sind.

Befehl >install
Terminal-Installierung
======================
Bitte rufen Sie Install jedesmal auf, wenn sie mit einem neuen Terminal
oder Terminalprogramm das Panther Mailbox System benutzen wollen.

Die folgenden 6 Fragen sollten sorgf}ltig von Ihnen werden, da Sie sonst
Schwierigkeiten mit der ]bertragung bekommen k/nnten. Das Ergebnis der
Befragung wird zusammen mit Ihrer ID abgespeichert und in Zukunft
automatisch nach dem Login abgerufen.ABCDEFGHIJKL
ABCD   EFGHIJK L

Sind diese beiden Zeilen auf Ihrem Bildschirm identisch? N

Erklingt nun ein Signal bei Ihnen?N

1.: ABCD
2.: ABCD
3.: ABCD
4.: ABCD

Bitte geben Sie die Nummer der ersten Zeile an, in der NUR AB steht:3

Dies ist die erste Zeile,und dies hier ist die Zweite.
Hat die zweite Zeile die erste Zeile ueberschrieben?J

Testzeichen: {|}[\]~
Haben Sie diese Zeichen als Umlaute empfangen?N

The quick brown fox jumped over the lazy dog.

The quick brown fox jumped over the lazy dog.
Sind beide Zeilen identisch?J

In Zukunft wird das PANTHER Mailboxsystem die eben gewaehlten
Parameter sofort nach der Identifikation automatisch einstellen.
```

Die Pantherbox präsentiert sich dem Neuling mit einem userfreundlichen Install-Menü

VON SEBASTIAN DOSCH



DIE ZEICHEN-KISTE

Teil 4

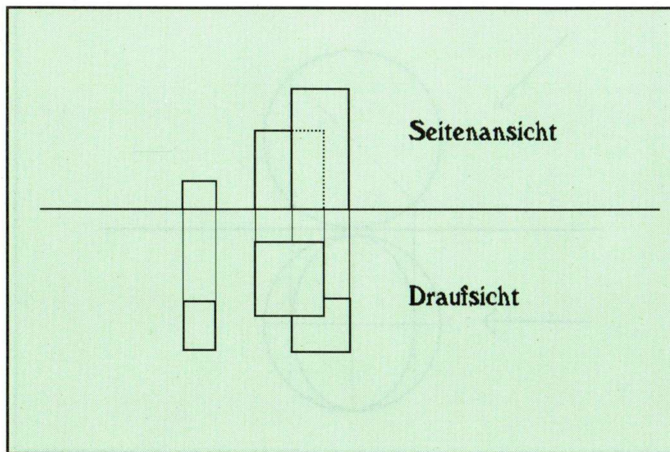
Schattenkonstruktion

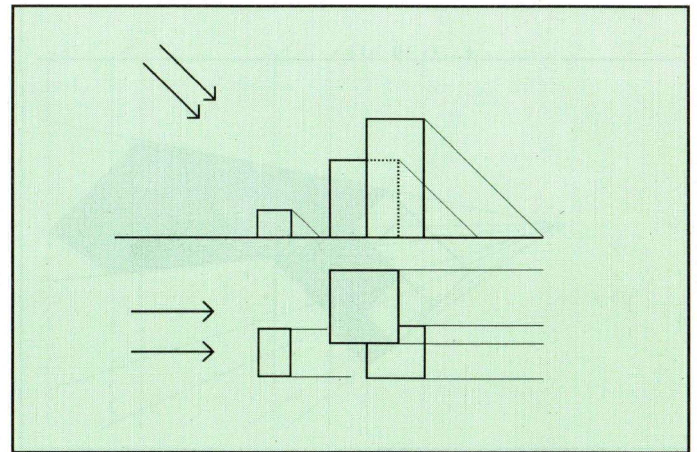
Heutiges Thema der Zeichenkiste soll die Konstruktion von Schatten sein. Daß dafür Licht der ausschlaggebende Faktor ist, besagt schon eine alte Volksweisheit: "Wo viel Licht, da auch viel Schatten". Wir werden diesem Leitfaden folgen.

Bei den Lichtquellen wird zwischen zwei Arten unterschieden: solchen, die Lichtstrahlen von einem Punkt aussenden (eine Lampe usw.), und solchen, die parallele Lichtstrahlen abstrahlen. Für welches System der Schattenbildung Sie sich entscheiden, hängt ganz von Ihnen ab.

Ich werde Ihnen nun die beiden verschiedenen Verfahren erklären, und möchte mit dem Schatten der parallel einfallenden Lichtstrahlen beginnen. Zuerst sollten Sie das "zu beschattende" Objekt im Grund- und Seitenriß zeichnen. Der Seitenriß wird am besten in der oberen Hälfte des Bildschirmes gezeichnet. Als zweckmäßig hat sich erwiesen, mit dem Grundriß zu beginnen, und dann die Ecken für die Seitenansicht auf diese zu projizieren. Wenden wir uns erst einmal der Draufsicht des Objektes zu. Zuerst werden die Lichtstrahlen und deren Richtung eingezeichnet. In der Seitenansicht folgt dann der Winkel, in dem die Strahlen einfallen sollen. Sie müssen danach in der Seitenansicht von den oberen Eckpunkten aus Linien in den Winkel Ihrer Lichtstrahlen zu der Grundlinie des Objektes ziehen. Achten Sie darauf, daß Sie wirklich nur die Eckpunkte mit der Grundlinie verbinden, die von Schatten betroffen sind. Danach projizieren Sie den Schatten der Seitenansicht auf die Draufsicht, wobei die Richtung der Lichtstrahlen zu beachten ist. Am einfachsten ist es, beim Grundriß von den Eckpunkten aus Hilfslinien in Richtung der Lichtstrahlen zu zeichnen, bis diese die Verbindung zwischen den Ecken und der Grundlinie aus der Seitenansicht schneiden (siehe Bild). Damit wäre der Schatten schon konstruiert. Sie sollten ihm noch eine dunklere Farbe geben, damit er sich von den anderen Teilen abhebt und eine realistischere Wirkung bekommmt. Abschließend werden die Konstruktionslinien entfernt.

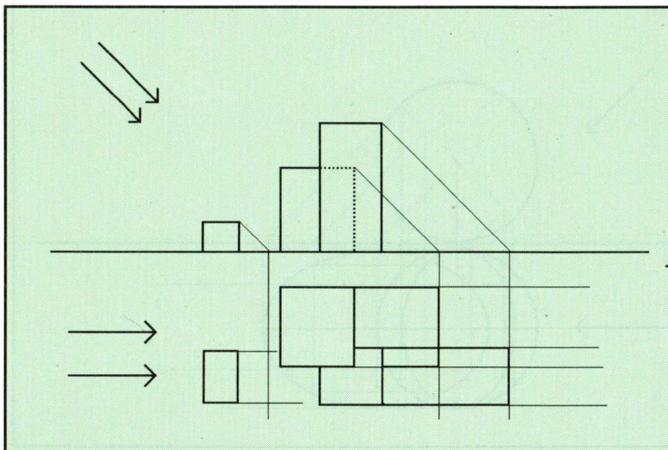
Kommen wir jetzt zu der zweiten Methode, der Konstruktion des Schattens bei einer punktförmigen Lichtquelle. Hierzu müssen Sie wieder das Objekt in der Seiten- und Draufsicht zeichnen. Projizieren Sie hierzu wieder die Eckpunkte aus dem Grundriß in die Seitenansicht. Zeichnen Sie nun die Lichtquel-

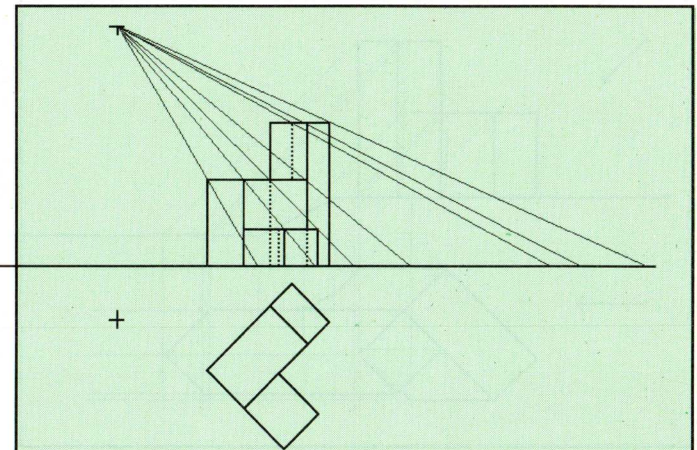


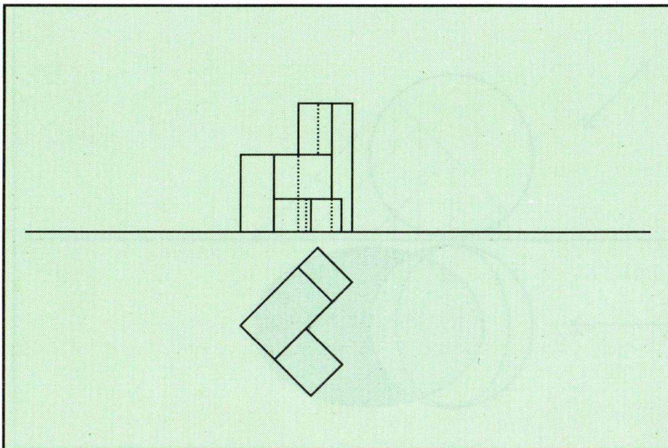


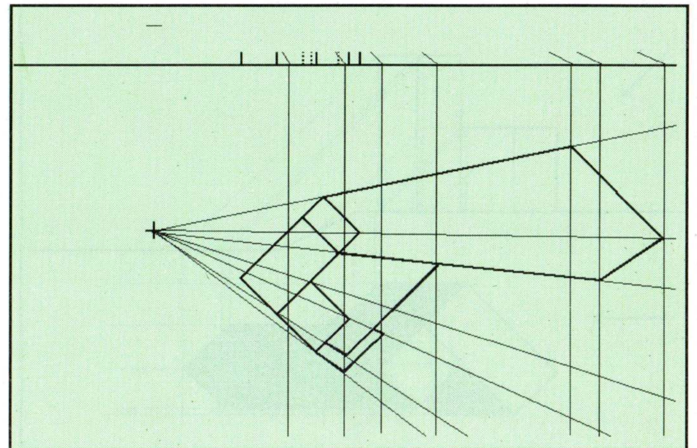
le in das Bild ein. Beachten Sie hierbei, daß eine in der Nähe des Bodens plazierte Lichtquelle einen sehr weiten und breiten Schatten bewirkt, während eine sehr hoch aufgesetzte Lichtquelle nur einen kurzen Schatten wirft. (Das gilt natürlich auch für die Schattenkonstruktion bei parallel einfallenden Lichtstrahlen.) Nachdem Sie die Lichtquelle in der Seitenansicht plaziert haben, sollten Sie diese auf die Draufsicht übertragen, um dadurch deren genauen Standort festzulegen. Ziehen Sie jetzt in der Seitenansicht Fluchtlinien von der Lichtquelle aus durch die Ecken des Objektes auf die Grundlinie. An den Schnittpunkten mit der Grundlinie zeichnen Sie senkrechte Geraden nach unten, bis hin zu der Draufsicht. Konstruieren Sie danach Hilfslinien in der Draufsicht von der Lichtquelle aus durch die Ecken des Objektes, bis sie die senkrechten Geraden aus der Seitenansicht schneiden. So erhalten Sie die Ecken Ihres Objektschattens mit punktförmiger Lichtquelle. Benutzen Sie jetzt nur noch die Hilfslinien in der Draufsicht als Schattenkanten, füllen Sie den Schatten dunkel aus, entfernen Sie die Hilfslinien und Sie sind fertig.

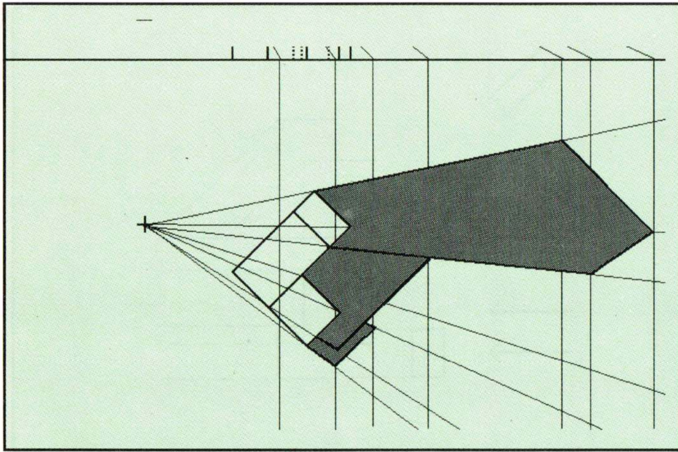
Aber nun möchte ich noch ein paar Sonderformen beschreiben. Nehmen wir z.B. eine Kugel. Da hier ja nicht mehr mit geraden Kanten, Ecken usw. gearbeitet werden kann, müssen wir eine große Anzahl von Hilfspunkten verwenden. Zuerst aber sollten wir ermitteln, ab welchem Punkt die Kugel Schatten wirft, und ab welcher Stelle sie selbst Schattenstellen aufweist. Der Einfachheit halber nehme ich hier die Konstruktion bei parallel einfallenden Lichtstrahlen. Halbieren Sie die Kugel mit einer Hilfsgeraden, die im rechten Winkel zu Ihren Lichtstrahlen verlaufen muß. Dies ist die Schattengerade auf der Kugel. Zeichnen Sie nun zwei Geraden von den obersten und den untersten Schattenkanten zur Draufsicht. Sie müssen hier eine Ellipse zeichnen, welche die obere und untere Außenkante der Kugel und

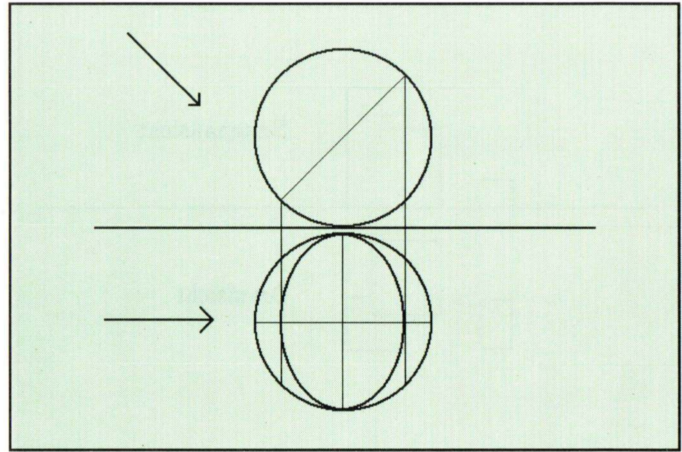




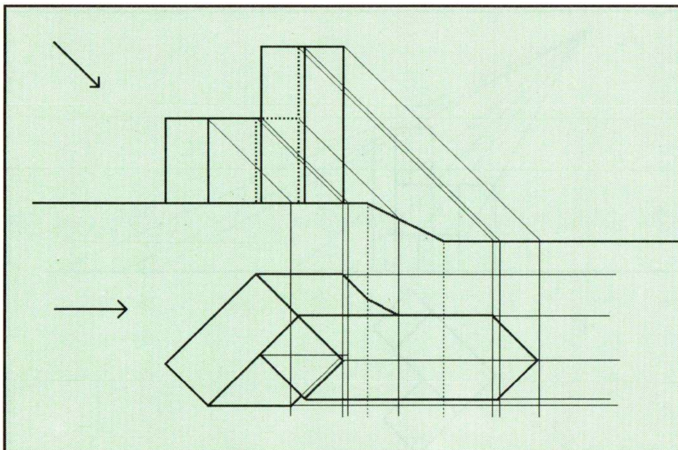


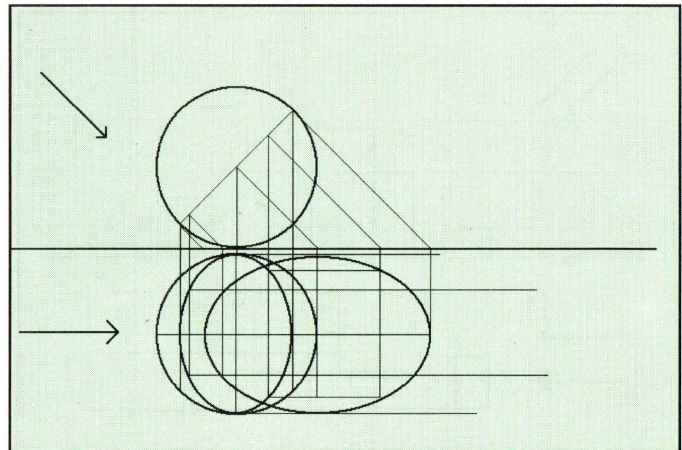






die beiden Hilfslinien schneidet und einen gemeinsamen Mittelpunkt mit der Kugel besitzt. Dadurch haben Sie die Schattenkanten auf der Kugel in der Draufsicht. Um nun den Schatten der Kugel zu zeichnen, müssen Sie in der nach unten, bis diese die untere Kante der Ellipse, die sich in der Kugel der Draufsicht befindet, schneiden. Von hier aus müssen nun waagerechte Linien zu den Geraden, die von der Grundlinie aus der Seitensicht kommen, gezeichnet in der Seitenansicht die Neigung oder die Steigung des Untergrundes einzeichnen, von den Eckpunkten aus auch auf die Grundebene Hilfslinien ziehen und das dann auf die Draufsicht übertragen (siehe Bild). Wie Ihnen wahrschein-





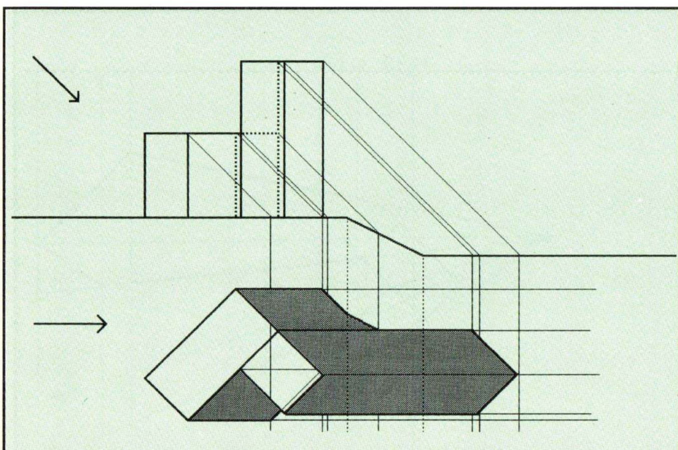
Seitenansicht von der Schattenkante aus Hilfslinien zeichnen. Ziehen Sie diese senkrecht nach unten zu der Draufsicht, von der Stelle aus, an der Sie die Grundlinie schneiden. Um die Ellipse des Schlagschattens (das ist der Schatten, der durch das Objekt auf dem Boden abgebildet wird) zu konstruieren, zeichnen Sie in der Seitenansicht an den Schnittstellen der Hilfslinien mit der Schattenkante auf der Kugel Geraden senkrecht

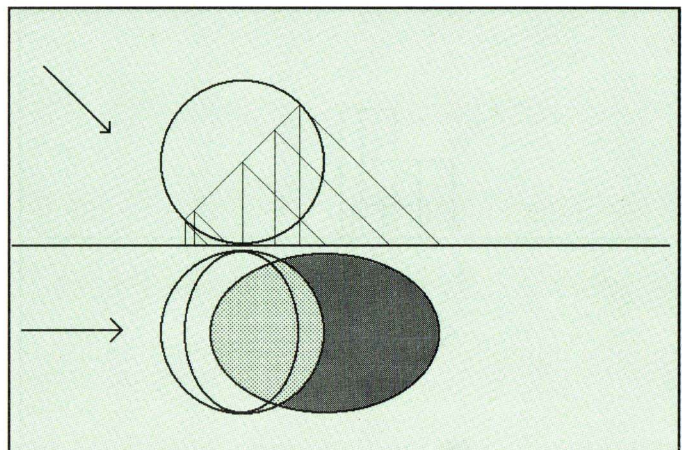
werden. Wenn Sie eine Elipse um alle diese Schnittpunkte auf der Grundebene zeichen, haben Sie den fertigen Schlagschatten der Kugel. Damit wären die wichtigsten Voraussetzungen geschaffen, um die Schattenkonstruktion der verschiedensten Objekte durchzuführen.

Für den Schatten bei geneigter oder aufsteigender Bildebene müssen Sie nur

lich schon aufgefallen ist, haben wir den Schatten diesmal nur in der Drauf- und in der Seitenansicht konstruiert. In der nächsten Ausgabe wird es dann darum gehen, wie Sie so komplexe Objekte wie Häuser, Rundbögen, Objekte mit Schatten usw. in der Schrägperspektive darstellen können

Also bis zum nächsten Mal, und viel Spaß beim Zeichen!





VON MARTIN SILBERNAGEL

Ein Amigavierundzwanzignadel-proportionaldruckertreiber oder: ein waaas?

Vielleicht gehören Sie, lieber Leser, ebenso wie ich auch zu jener Sorte von paranoiden Freaks, die sich einen dieser sauteuren 24-Nadler gekauft haben, weil der Probeausdruck beim Händler so toll war den aber die Ausdrucke, die der AMIGA so über die Schnittstelle schickte, später auf den Lampenschirm brachten, weil der Blocksatz eher wie eine Mischung zwischen Käsekuchen und Lochstreifen aussah?

Um dieser dummen Situation zu entkommen, sah ich seinerzeit vier Wege:

alpha: Ich werfe meinen Drucker aus dem Fenster.

beta: Ich werfe meinen AMIGA aus dem Fenster.

gamma: Ich springe aus dem Fenster.

delta: Ich klemme mich hinter meinen C-Compiler.

Als die Lösungen *alpha* bis *gamma* versagten, weil mein Zimmer im Erdgeschoß liegt, resignierte ich und wollte dann die vierte Lösung realisieren.

Weil wohl wieder wahrlich widrige Winde wehten, verzog ich mich in eine ruhige Ecke meines Stuhls und dachte nach, was zu tun sei, um die schon eingebauten Leistungen meines Druckers auszuschöpfen, die bis jetzt immer brachlagen. Ich kam dabei zu dem Schluß, daß Textprogramme auf dem AMIGA bis jetzt nur auf zwei Arten funktionieren bzw. ausdrucken:

aaah: Der Text wird zeilenweise an den Drucker ausgegeben. Das bedeutet, daß die Zeilen alle buchstabenweise formatiert werden, also in ihre Grenzen gewiesen und an ihre Position gebracht werden. Das funktioniert alles wunderbar, solange man den Drucker nicht auf Proportionalschrift bringt, wo das kleine "i" viel weniger Platz braucht als das große "W". Dann nämlich rasten die meisten Textprogramme aus. Um diesem Problem aus dem Weg zu gehen, haben sich ein paar ganz schlaue Programmierer einen anderen Weg einfallen lassen:

`U/`u	Unterstreichen an/aus
`F/`f	**Fettdruck** an/aus
`D/`d	**Doppeldruck** an/aus
`H/`h	Hochstellen an/aus
`T/`t	Tiefstellen an/aus
`S/`s	Schmaldruck an/aus
`B/`b	Breitdruck an/aus
`K/`k	*Kursivdruck* an/aus
`P/`p	Proportionaldruck an/aus
`0	Linksbündig
`1	Mittig
`2	Rechtsbündig
`3	Blocksatz
`A	3 Zeilen pro Zoll
`4	4 Zeilen pro Zoll
`6	6 Zeilen pro Zoll
`8	8 Zeilen pro Zoll
`!	Tabulator
`@	Seitenvorschub

Zeitloses Dokument Nummer 4

beeh: Der Text wird als HiRes Grafik ausgegeben. In der Praxis bedeutet das bei den Kickstart 1.2-Druckertreibern bis zu 20 Minuten Wartezeit pro Seite. Um den Herrn der Ringe auszudrucken bräuchte man also knapp 20 Tage. Doch das ist nicht der einzige Nachteil. Da die Druckroutine maximal 1024 Punkte pro Zeile verwalten kann, werden diese auf die 1920 Punkte (bei einem normalem 24-Nadler) verteilt. Dadurch sind Verzerrungen und ein Schriftbild, schlimmer als die Handschrift von Olaf Przybilla (Wer kennt ihn nicht?) eine logische Konsequenz, die durch die Verwendung von Bildschirmzeichensätzen unterstützt wird. Denn diese Bilschirmzeichensätze heißen so, weil sie auf Druckern wirklich absolut bedröselt aussehen.

Was ist also zu tun? Würden Sie das Handbuch Ihres 24-Nadlers mal genauer angucken, so hätten Sie wahrscheinlich festgestellt, daß der Drucker selber über meh-

rere leistungsfähige Befehle verfügt, um den Text zu formatieren und im Aussehen zu variieren. Nach dem Anschalten formatiert der Drucker die Texte zwar nur links und druckt alle Zeichen gleich breit, senden Sie aber die Zeichenkette "esc'p'1", schalten Sie damit den Proportionalzeichensatz ein (siehe dazu *zeitloses Dokument Nummer eins*). Nun haben Sie ein Schriftbild, das schon eher dem entspricht, was Hausfrauen wünschen.

Wenn Sie nun noch die Returns hinter den Zeilen weglassen (Fließtext nennt man das, hat Markus gesagt, wenn außer am Ende des Absatzes kein Return mehr vorkommt), werden Sie sehen, daß Ihr Drucker automatisch einen Wordwrap durchführt. Auf Germanisch bedeutet das, daß das (gut, nä; gleich drei 'das' hintereinander. Hab' lange rumgeknobelt, bis ich das hatte) Wort am Ende der Zeile in die nächste kommt, wenn kein Platz mehr da ist (siehe dazu *zeitloses Dokument Nummer zwei*).

Als Höhepunkt könnten Sie jetzt noch die Formatierungsart von linksbündig auf mittig, rechtsbündig oder Blocksatz stellen, und Sie würden feststellen, daß auch hier der Drucker automatisch die Zeilen formatiert, und erst bei einem Return einen Absatz macht (siehe dazu *zeitloses Dokument Nummer drei*).

Zur Verschönerung des Textes könnte man nun dazu übergehen, per esc-Kommandos zu unterstreichen, und wäre wieder mal drauf und dran, durchzudrehen. Denn versuchen Sie mal, in einen Text direkt esc-Kommandos bzw.-Sequencen einzugeben. Ich wette, daß Sie schnell wieder davon abkommen werden.

Und genau an diesem Punkt setzt mein Druckertreiber an. Er liest einen Text ein und sendet esc-Kommandos, sobald er auf das Zeichen "`" (oberhalb der Tab-Taste) und einen entsprechenden Kommandobuchstaben stößt. Genial, was? (siehe dazu *zeitloses Dokument Nummer vier*).

Das glauben Sie mir nicht, daß der das kann? Da lach' ich doch: ha, ha!
Geben Sie das Programm doch einfach selber ein und lassen es compilieren. Ich würde Ihnen zum Aztec V3.6a raten, weil ich das Programm unter dem entwickelt hab'. Aber Sie können von mir aus auch sonstwas verwenden (falls Sie den Lattice verwenden, fällt die Anweisung "*include<functions.h>*" natürlich weg).
Falls Sie allerdings doch mit dem Aztec compilieren, gehen Sie am besten so vor: Erst "*ccprop*" und dann "*lnproplc*" eingeben. Dann sehen Sie selbst, daß mein Programm das kann.
Außerdem kann es, wenn in der Command-Line die Option EDITOR gesetzt ist, Editortexte (also Return hinter jeder Zeile) in Fließtexte übersetzen. Das macht es ganz einfach, indem es alle

```
Vorher: AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz
Nacher:  AaBbCcDdEeFfGgHhIiJjKkLlMmNnOoPpQqRrSsTtUuVvWwXxYyZz
```

Zeitloses Dokument Nummer 1

```
Hallo, ihr lieben und treuen Leser und ihr Leute zuhause an oiren Großrechnern und AMIGAS
... have fun          (Return)
```

Zeitloses Dokument Nummer 2

```
        Hallo, ihr lieben und treuen Leser und ihr        (Return)
            Leute zuhause an oiren Großrechnern   (Return)
                und AMIGAS... have fun        (Return)
```

Zeitloses Dokument Nummer 3

Selbstportrait

Auf seinem Tisch stand einen Packung mit geschenkten Pralinen, die seltsamerweise, obwohl solche Dinge wie 'Cerise-Trüffel' oder 'Arak-Zartkrokant' darauf standen, alle irgendwie nach Schaumgummi schmeckten. Er aß sie trotzdem.
Denn beim Essen kann man bestens philosophieren: über's Programmieren zum Beispiel. Er kam zu dem Schluß, daß er normalerweise solange an seinem Computer saß, bis entweder sein Programm (dank DOS) für immer und ewig verschwunden war und er erleichtert ins Bett ging - oder bis er soviele *BUGS* in seinem Programm hatte, bis es gar nicht mehr lief. Dann ging er für gewöhnlich nicht so erleichtert ins Bett.

Doch plötzlich wurde er von zwei Dingen in die Realität geholt: erstens war sein Compiler endlich fertig geworden und zweitens war ihm plötzlich ganz fürchterlich übel geworden (Wovon bloß?).
Wie zu erwarten lief das Programm nicht, aber der Absturz den das Programm produzierte, war sehenswert. Er war mit sich zufrieden und speicherte das Programm auf Festplatte.

Daß er in Zeitdruck war störte ihn eigentlich nicht weiter, denn er wußte, daß er nicht mehr fertig werden würde. Warum sich also Aufregen. Logo™, oder?

```
`L00`R70`1`U`BSelbstportrait`u`b`

`3`BA`buf seinem Tisch stand einen Packung mit geschenkten Pralinen, die
 seltsamerweise, obwohl solche Dinge wie ``Cerise-Trüffel' oder
 ``Arak-Zartkrokant' darauf standen, alle irgendwie nach Schaumgummi
 schmeckten. Er aß sie trotzdem.`
Denn beim Essen kann man bestens philosophieren: über's Programmieren zum
 Beispiel. Er kam zu dem Schluß, daß er normalerweise solange an seinem
 Computer saß, bis entweder sein Programm (dank DOS) für immer und ewig
 verschwunden war und er erleichtert ins Bett ging - oder bis er soviele
 BUGS in seinem Programm hatte, bis es gar nicht mehr lief. Dann ging er
 für gewöhnlich nicht so erleichtert ins Bett.`

`BD`boch plötzlich wurde er von zwei Dingen in die Realität geholt: erstens
 war sein Compiler endlich fertig geworden und zweitens war ihm plötzlich
 ganz fürchterlich übel geworden (Wovon bloß?).`
Wie zu erwarten lief das Programm nicht, aber der Absturz den das Programm
 produzierte, war sehenswert. Er war mit sich zufrieden und speicherte das
 Programm auf Festplatte.`
Daß er in Zeitdruck war störte ihn eigentlich nicht weiter, denn er wußte,
 daß er nicht mehr fertig werden würde. Warum sich also Aufregen.
 Logo`Htm`h, oder?`
```

Zeitloses Dokument Nummer 5

Returns überliest, es sei denn, vor dem Return kommt ein (was denn sonst) "!"-Kommandozeichen. Das ist ziemlich wichtig, denn fast kein Textprogramm (außer der Betarelease von Textcraft-Plus) kann Fließtext erzeugen.

Aber wenn wir grad' bei den Optionen sind: durch die Option SHOW gibt der Treiber den Text ohne "!"-Kommandos in den "*stdout*"-Kanal aus. Dieser Kanal endet normalerweise im aktuellen CLI-Fenster, in dem Sie dann den Text mitverfolgen können (wenn z.B. der Drucker 10km entfernt steht).

Damit Sie besser hinter die ganze Sache steigen, hab ich für Sie einen kleinen Probetext geschrieben, den Sie einmal so sehen, wie er im Editor eingegeben wird, und einmal, wie ihn der Drucker erzeugt. Wenden Sie also Ihren Blick zum *zeitlosen Dokument Nummer fünf.* Nun nehmen Sie sich irgendeinen Editor (wenn's geht, am besten den KICK-ED) und geben den Text so ein, wie Sie ihn sehen. Dann speichern Sie ihn irgendwohin ab, aber merken sich bitte den Pfadnamen (wär' toll). Dann verlassen Sie Ihren Editor, so schwer Abschied nun mal fällt, und geben im CLI folgendes ein:

prop Pfadname EDITOR SHOW

Nun sollten Sie (sofern noch genügend Farbe im Band ist) das supertolle Ergebnis sehen können.

Und falls das funktioniert hat, könnten Sie sich das Programm ja mal genauer anschauen: Z.B benutzt es, um den Commo-Druckertreiber zu umgehen, als Zielfile PAR:. Wenn Ihre Kreissäge allerdings an der seriellen Schnittstelle hängt, muß stattdessen SER: stehen (na chlor).

Weiterhin wichtig wär vielleicht noch die Translationstabelle (tolles Fremdwort). Mit dieser Tabelle werden die AMIGA-Zeichencodes in Druckercodes umgewandelt. Wichtig ist das z.B. für die Zeichen "à", "œ" oder "¨Ω".

Der Rest vom Programm erklärt sich eigentlich selbst, find' ich. Erstellt wurde es für einen EPSON LQ800, haben Sie jedoch einen anderen Drucker, so kann nicht ausgeschlossen werden, daß Sie abweichende Steuercodes eingeben müssen. Da hilft aber ein Blick ins

```
/*******************************************************************/
/*                                                                 */
/* Proportionaldruckertreiber für alle 24-Nadler mit ESC/P Kommandos */
/*                                                                 */
/* V1.3 geschrieben am 15/01/89 von Martin Silbernagl              */
/*                                                                 */
/*                (c) 1988 by Martin Silbernagl                    */
/*                                                                 */
/*******************************************************************/

#include <functions.h>                  /* hab Aztec V3.6a benutzt  */
#include <libraries/dos.h>
#define  esc 27                         /* der Übersicht wegen      */
#define  NUM_OF_COMS      29            /* Anzahl der Kommandos     */

USHORT   *file_in,*file_out;            /* file-pointer in/out      */
USHORT   i;                             /*     Zählvariable         */
UBYTE             mychar,lastchar;      /* Lesebuffer               */
UBYTE             what;                 /* Lesebuffer               */
UBYTE             ed_file,show;         /* Flags*/

UBYTE    Initstring[] =                 /* Drucker-Initalisierung   */
{
 esc,'@', esc,'x',1, esc,'p',1, esc,'a',3, esc,'t',1
};

UBYTE Coms[] =                          /* Kommandobuchstaben       */
{
 'U','u',          /* Unterstreichen ein/aus */
 'F','f',          /* Fettdruck      ein/aus */
 'D','d',          /* Doppeldruck    ein/aus */
 'H','h',          /* Hochstellung   ein/aus */
 'T','t',          /* Tiefstellung   ein/aus */
 'S','s',          /* Schmaldruck    ein/aus */
 'B','b',          /* Breitdruck     ein/aus */
 'K','k',          /* Kursivdruck    ein/aus */
 'P','p',          /* Proportional   ein/aus */
 '0','1',          /* linksbündig / mittig   */
 '2','3',          /* rechtsbündig / block   */
 'A','4',          /* 3 bzw. 4 Lines per Inch */
 '6','8',          /* 6 bzw. 8 Lines per Inch */
 '!','@',          /* Tabulator / Seitenende  */
 '\n'              /*     Return              */
};

UBYTE Seq[] =      /* 3 Byte Kommandos        */
{
   esc,'-',1,          esc,'-',0,
   esc,'E',NULL,       esc,'F',NULL,
   esc,'G',NULL,       esc,'H',NULL,
   esc,'S',0,          esc,'T',NULL,
   esc,'S',1,          esc,'T',NULL,
   esc,15,NULL,        18,NULL,NULL,
   esc,'W',1,          esc,'W',0,
   esc,'4',NULL,       esc,'5',NULL,
   esc,'p',1,          esc,'p',0,
   esc,'a',0,          esc,'a',1,
   esc,'a',2,          esc,'a',3,
 esc,'3',60,                       esc,'3',45,
   esc,'0',NULL,       esc,'2',NULL,
   9,NULL,NULL,                12,NULL,NULL,
 10,0,0
};

UBYTE LeftBorder[] =          /* setzt linken Rand   */
{
 esc,'l',0
};

UBYTE RightBorder[] =         /* setzt rechten Rand  */
{
 esc,'Q',0
};

UBYTE     Translation[] =
{
```

Druckerhandbuch, was nicht schwerfallen dürfte, es sei denn Sie haben nur eine Raubkopie des Druckers und deshalb keine Dokumentation dazu (kleiner Scherz zum Ende!).

Schönen Tag auch noch.

Keine schönen Grüße an die 11a.

Euer Martin Friedrich Ludwig Edler Freiherr von Silbernagl.

```
        0,  1,  2,  3,  4,  5,  6,  7,
        8,  9, 10, 11, 12, 13, 14, 15,
       16, 17, 18, 19, 20, 21, 22, 23,
       24, 25, 26, 27, 28, 29, 30, 31,
       32, 33, 34, 35, 36, 37, 38, 39,
       40, 41, 42, 43, 44, 45, 46, 47,
       48, 49, 50, 51, 52, 53, 54, 55,
       56, 57, 58, 59, 60, 61, 62, 63,
       64, 65, 66, 67, 68, 69, 70, 71,
       72, 73, 74, 75, 76, 77, 78, 79,
       80, 81, 82, 83, 84, 85, 86, 87,
       88, 89, 90, 91, 92, 93, 94, 95,
       96, 97, 98, 99,100,101,102,103,
      104,105,106,107,108,109,110,111,
      112,113,114,115,116,117,118,119,
      120,121,122,123,124,125,126,176,
        0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,
        0,  0,  0,  0,  0,  0,  0,  0,
       32,173,155,156,249,157,124,21,
       34,99,166,174, 170,196, 82,169,
      248,241,253,252,251,230,227,250,
       44,254,167,175,172,171, 32,168,
      133,160,131,32,142,143,146,128,
      138,144,136,137,141,161,140,139,
       32,165,149,162,147, 32,153,  0,
       48,151,163,150,154, 32, 32,225,
      133,160,131,32,132,134,145,135,
      138,130,136, 137,141,161,140,139,
       32,164,149,162,147, 32,148,  0,
       32,151,163,150,129, 32, 32,152
  };

  main(argc,argv)
  int argc;
  char *argv[];
  {
     if(argc<2)          /* keine Argumente? */
     {
        info:
        printf("Use: 1> ");
        printf(argv[0]);
        puts(" inputfile [EDITOR] [SHOW]");
        exit(RETURN_FAIL);
     }
   puts("(c) 15/01/1989 by Martin Silbernagl\n");
     ed_file=FALSE;      /* Fließtext         */
     show=FALSE;         /* nicht mitschreiben */
     for(i=2; i<argc; i++)
     {
        if(strcmp("EDITOR",argv[i])==0)
        ed_file=TRUE;
        else if(strcmp("SHOW",argv[i])==0)
        show=TRUE;
        else
        {
           puts("bad args!");
           goto info;
        }
     }
   if((file_in=Open(argv[1],MODE_OLDFILE))==NULL)
                                  /* Input öffnen    */
     {
        puts("Unable to open input file");
        exit(ERROR_OBJECT_NOT_FOUND);
     }
   if((file_out=Open("par:",MODE_NEWFILE))==NULL)
                                  /* Drucker öffnen*/
     {
        puts("Unable to open PAR: for output");
        exit(ERROR_OBJECT_NOT_FOUND);
     }
   Write(file_out,Initstring,14L);
                      /* Drucker intialisieren */
   while(Read(file_in,&mychar,1L)!=0)
                          /* ein Zeichen holen */
   {
      if(ed_file) TestRet();    /* Fließtext? */
      if(mychar=='|') DoThings();
                                  /* Kommando?*/
      if(show) printf("%c",mychar);
                               /* Mitschreiben */
         mychar=Translation[mychar];
                        /* in ASCII umwandeln  */
      Write(file_out,&mychar,1L);
                           /* Zeichen ausgeben */
   }

   Close(file_in);
   Close(file_out);
   return(RETURN_OK);
}

TestRet()                         /* Return? */
{
   if(mychar==10) mychar=0;
   return(0L);
}

DoThings()           /* Kommandos ausgeben*/
{
   Read(file_in,&what,1L);
   if(what=='|') return(0L);

   for(i=0; i<NUM_OF_COMS; i++)
   {
      if(what==Coms[i])
      {
         Write(file_out,&Seq[i*3],3L);
            goto end_do;
      }
   }

  if(what=='L' || what=='l')
  {
         LeftBorder[2]=GetBorder();
         Write(file_out,LeftBorder,3L);
         goto end_do;
  }
  if(what=='R' || what=='r')
  {
         RightBorder[2]=GetBorder();
         Write(file_out,RightBorder,3L);
         goto end_do;
  }

  end_do:
   mychar=0;
   return(0L);
}

GetBorder()      /* Zweistellige Ziffernfolge
                               umwandeln */
{
  UBYTE    buffer;

  Read(file_in,&what,1L);
  if(what<'0' || what >'9') return(0);
  what-='0';
  what*=10;
  buffer=what;
  Read(file_in,&what,1L);
  if(what<'0' || what >'9') return(0);
  what-='0';
  what+=buffer;

  return(what);
}
```

ENDE

VON THOMAS PFOCH

Das Acht-Damen-Problem

Dieser Artikel soll einen Vergleich der konventionellen, optimierten Programmierung mit den KI-Sprachen ermöglichen. Das soll keinesfalls den Nutzen der KI-Sprachen bestreiten, es soll vielmehr zeigen, wie groß die Leistung herkömmlicher Programmiersprachen für klar umrissene Probleme ist.

Des weiteren soll an einem einfachen Beispiel gezeigt werden, wie die Umsetzung eines Problems in ein das Problem lösendes Programm erfolgt. Es wird das Prinzip der Rekursion erläutert und ein kurzer Einblick in Bitoperationen bei der Programmiersprache 'C' gegeben.

Problemdefinition

Das Acht-Damen-Problem beschäftigt sich, wie der Name schon sagt, mit acht Damen (Schachfiguren). Diese sollen so auf einem Schachbrett positioniert werden, daß sie sich gegenseitig nicht schlagen können. Die Bewegungsmöglichkeiten sind die einer Dame im Schachspiel (senkrecht, waagerecht und diagonal).

Lösungsansatz

Unser Programm könnte jetzt natürlich einfach alle acht Damen über das gesamte Schachbrett laufen lassen und abprüfen, ob die Damen erstens nicht auf identischen Positionen stehen und zweitens, daß sie sich gegenseitig nicht schlagen können.

Dieses Programm besteht aus acht ineinander geschachtelten Schleifen, die je einer Dame zugeordnet sind und diese über alle 64 Felder laufen lassen. Der damit verbundene Zeitaufwand liegt bei unserem Rechner bei etwa 7000 Jahren (optimistisch); ein wohl offensichtlich indiskutabler Zeitraum. Es obliegt also das Problem der Optimierung.

Optimierung

Wenn man sich an ein Schachbrett setzt und versucht, die Damen darauf zu verteilen, stellt man schnell fest, daß es sinnlos ist, mehr als eine Dame in einer Reihe aufzustellen.
Diese Erkenntnis nutzen wir, indem wir unser Schachbrett nicht etwa als Matrix mit acht mal acht Plätzen vereinbaren, sondern für jede Reihe nur ein Feldelement benutzen; der Wert, der in dem Feldelement steht, gibt uns hierbei die Position der Dame in der Reihe an (s. Abb. 1).

Unsere Schleifen müssen nun nicht mehr über das gesamte Spielfeld laufen, sondern nur noch über die der Dame zugehörige Reihe (also nicht von 1 bis 64, sondern von eins bis acht). Der Zeitaufwand verkürzt sich damit auf ca. 1600 Sekunden (wie oben wird von 10.000 Stellungsüberprüfungen pro Sekunde ausgegangen).

Eine weitere Möglichkeit der Optimierung ergibt sich, wenn man sich das Setzen der Damen in verschiedenen Reihen auf dem Schachbrett vorstellt. Was für die Reihen gilt, gilt in gleicher Weise auch für die Spalten; es darf nur jeweils eine Dame gesetzt werden (die Überprüfung der Diagonalen folgt später).

Richten wir uns also eine Maske ein, die die bereits besetzten Spalten repräsentiert. Da ich als Implementationssprache 'C' gewählt habe, ist das auf eine sehr einfache Weise möglich: Man nimmt ein <*unsigned*> und setzt das der Spalte entsprechende Bit (s. Abb. 2).

Unser Programm prüft nun ganz einfach, ob in der Spalte, in die wir unsere Dame setzen wollen, das entsprechende Bit gesetzt ist. Ist dies der Fall, so brauchen wir uns um die tieferen Reihen nicht mehr zu kümmern (es gibt ab dieser Reihe sowieso immer zwei Damen, die sich schlagen können).

Jetzt sind wir an einem Punkt angekommen, an dem es sinnvoll ist, die Möglichkeit der rekursiven Programmierung (s. Abb. 3) zu nutzen.

Wir lassen unsere geschachtelten Schleifen das sein, was sie sind (schön und anschaulich), und begeben uns auf das Glatteis der Rekursion. Hierzu brauchen wir eine Funktion, deren Aufgabe es ist, eine Dame durch eine Reihe zu schieben (s. Abb. 4) und abzuprüfen, ob diese Dame von einer vorher plazierten geschlagen werden kann. Ist dies der Fall, so kann keine Kombination der Damen in den tieferliegenden Reihen eine Lösung ergeben. Kann die neu plazierte Dame jedoch nicht geschlagen werden, ruft sich die Funktion rekursiv (bzw. selbst) auf und erzeugt damit Damenstellungen in den tieferen Reihen. Der Ebenenparameter (vgl. Abb. 3) bei diesem Aufruf ist natürlich die tiefere Reihe. In ihr wird der beschriebene Vorgang wiederholt.

Im hierzu genutzten Datentyp *<unsigned>* (vgl. Maske), wird das Bit gesetzt, das die Position in der Reihe kennzeichnet.
Dies läßt sich am einfachsten erreichen, indem man ein Bit des *<unsigned>* setzt und dieses durch 'Shiften' über die einzelnen Positionen schiebt. Durch eine ODER-Verknüpfung von Maske und Damenposition läßt sich so leicht prüfen, ob die Dame in eine bereits besetzte Spalte gesetzt werden soll.

Natürlich muß, wie jede Rekursion, unsere Funktion auch ein Abbruchkriterium haben. Dieses ist erfüllt, wenn wir eine Dame in der letzten (achten) Reihe plaziert haben. In diesem Fall wurde eine Lösung gefunden. Statt eine weitere Rekursionsebene zu erzeugen, zählen wir den Zähler für die Lösungen um Eins hoch und fahren mit dem Verschieben der Dame fort.

Wenn eine Dame in der letzten Spalte angekommen ist, haben wir alle Positio-

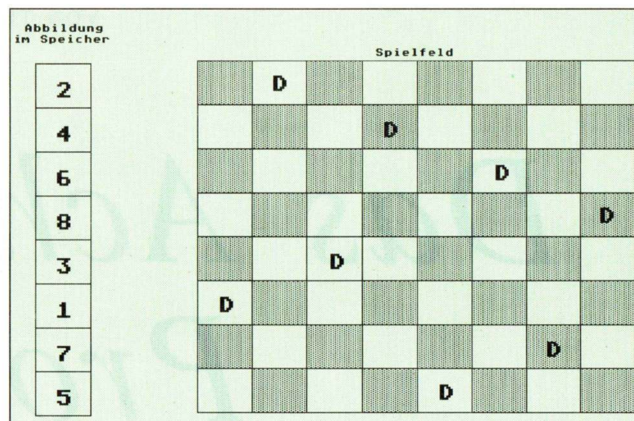


Abb. 1:
Die Damen bewegen sich jeweils nur in einer Reihe; für jede Reihe wird ein <unsigned> reserviert, unser Feld (acht Elemente) kann so das gesamte Spielfeld repräsentieren.
Die Werte in der Abbildung geben die Spalte an, in der sich die Dame befindet. Die Speicherrepräsentation geht aus Abb. 4 hervor.

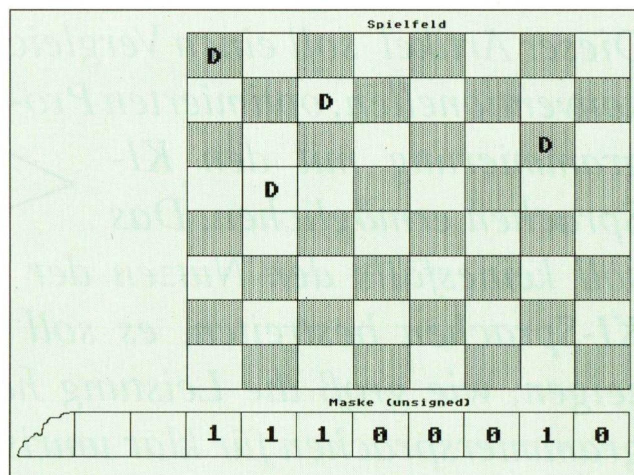


Abb. 2:
Unsere Maske repräsentiert die durch Damen besetzten Spalten (siehe auch Abb. 4).

Rekursive Funktionsaufrufe

```
Routine_A(Ebene) {
    IF (Ebene >= maximale Rekursionstiefe) Abbruch;
    ELSE Routine_A(Ebene+1);
} /* einseitige Rekursion */

Routine_1(Ebene) {
    IF (Ebene >= maximale Rekursionstiefe) Abbruch;
    ELSE Routine_2(Ebene);
} /* wechselseitige Rekursion, Funktion 1 */

Routine_2(Ebene) {
    Routine_1(Ebene+1);
} /* wechselseitige Rtekursion, Funktion 2 */
```

Abb. 3:
Rekursion bedeutet, daß eine Funktion sich selbst, evtl. über Zwischenaufrufe, aufruft. Das Prinzip ähnelt dem von geschachtelten Schleifen, hat jedoch den wesentlichen Vorteil, daß die Zahl der Schachtelungen nicht vorher festgelegt werden muß.
Die Anzahl der Schachtelungen wird durch die Rekursionstiefe bestimmt; außerdem werden in jeder Rekursionsebene die gleichen Operationen ausgeführt (nicht, wie bei geschachtelten Schleifen, nur in der innersten). Damit die Funktion sich nicht unendlich oft selbst aufruft, muß sie überprüfen, wie oft sie bereits geschachtelt wurde; dies wird durch die Übergabe eines Parameters erreicht, der die aktuelle Rekursionstiefe (in gewünschter Form) enthält. Bei jedem Selbstaufruf der Funktion sollte dieser Parameter verändert werden, damit sich die tiefergeschachtelte Funktion über ihre Schachtelungsebene im klaren ist.

Die Rekursion wird abgebrochen, wenn sie ihre maximale Tiefe erreicht hat. Ein Abbruch bedeutet, daß die weitere Schachtelung beendet wird, als nächstes also die vorher aktive (aufrufende, höhere) Funktion aktiviert wird. Wird die oberste Rekursionsebene verlassen, so ist die Rekursion beendet. Mit jedem Funktionsaufruf innerhalb der Rekursion ist ein Ebenenwechsel verbunden. Rekursion ist das grundlegende Prinzip der KI-Sprachen.

nierungsmöglichkeiten geprüft. Jetzt kann der Funktionsaufruf beendet und die Rekursion auf der tieferen Ebene fortgesetzt werden (d.h. wir schieben die Dame in einer höheren Reihe weiter).

Hat die Funktion der ersten Rekursionsebene (erste Reihe) die Dame auf das letzte Feld gesetzt und die zugehörigen Möglichkeiten überprüft, so wird auch sie terminieren. Die Rekursion ist dann vollständig abgearbeitet, und unser Zähler enthält die Anzahl der Möglichkeiten. Das Hauptprogramm wird wieder aktiviert und kann diese Zahl ausgeben.

Mit der Rekursion haben wir den wichtigen Effekt, daß für jede Ebene (Reihe) eine neue Maske angelegt wird; wir müssen also beim Aufstieg auf eine höhere Reihe nicht den alten Zustand wiederherstellen.

Wir haben durch die Rekursionsebenen sichergestellt, daß in jeder Reihe nur eine Dame stehen kann; die Maske garantiert uns, daß in jede Spalte nur eine Dame gesetzt wird; das letzte noch bleibende Problem ist das Überprüfen der Diagonalen.

Hierzu müssen wir prüfen, ob die zuletzt gesetzte Dame in einer Diagonalen mit den vorher gesetzten steht. Dazu kopieren wir uns einfach das Feldelement mit der Damenposition (wie aus Abb. 4 hervorgeht, kann man es als Maske für eine Reihe interpretieren) in zwei neue Masken, die mit rechts bzw. links bezeichnet werden. Diese beiden Masken werden pro Reihe, die wir nach oben laufen, jeweils um ein Bit nach links bzw. rechts verschoben und decken damit die Diagonalen zu der Dame ab (vgl. Abb. 5).

Das so optimierte Programm hat eine Laufzeit von ca. 0.6 Sekunden, es fehlt uns allerdings noch der Vergleich zu den Benchmark-Tests der KI-Sprachen. Diese wurden für ein sechs-mal-sechs-Schachbrett mit sechs Damen gemacht; das abgedruckte Programm (Listing 1) berechnet die Lösungszahl für Schachbretter der Kantenlänge eins bis fünfzehn. Für ein sechs-mal-sechs-Brett habe ich bei meinen Tests Laufzeiten von etwa 0.04 Sekunden gemessen, selbstverständlich ohne Ausgabe der

Abb. 4: Die Werte, die für eine Dame gespeichert werden, geben nicht etwa die Nummer der Spalte an, in der die Dame steht, sondern werden analog zu der in der Maske benutzten Repräsentation durch Bits benutzt.

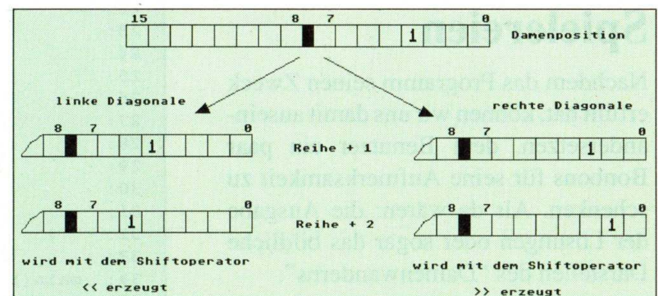


Abb. 5: Das Erzeugen der Diagonalen erfolgt, indem die Damenposition pro Reihe um eins nach links bzw. nach rechts geshiftet wird. Finden wir in einer der Diagonalen eine bereits gesetzte Dame, so können sich diese schlagen; es wird ein Fehlercode zurückgegeben. Sollten wir alle höheren Reihen überprüft haben, ohne auf eine Dame zu treffen, kann die neue Dame nicht geschlagen werden, es wird kein Fehlercode zurückgeliefert und der Rekursionsfunktion damit mitgeteilt, daß es sich lohnen kann, die tieferen Reihen (Rekursionsebenen) mit Damen zu besetzen.

```
#include <stdio.h>

#define besetzt 1
#define frei 0

int Feld[8];
long Anzahl;

Setze_Dame(Reihe,Maske)
  register int Reihe,Maske;
  { register int Spalte;

      for(Spalte=1;Spalte<=0xff;Spalte<<=1)
        { if((Maske&Spalte)==0)        /*ist spalte frei*/
            { Feld[Reihe]=Spalte;/*merken */
              if(diagonal(Reihe)==frei)
                { if(Reihe>=7) Anzahl++;/*loesung*/
                  else Setze_Dame(Reihe+1, (Maske|Spalte));
                }
            }
        }
  }

diagonal(Reihe)
  register int Reihe;
  {  register int i,*Testreihe,links,rechts;
     Testreihe=Feld+Reihe;
     links=rechts=*Testreihe--;
     for(i=1;i<=Reihe;i++,Testreihe--)
               /*rückwärts über alle reihen*/
       { links<<=1;    /*diagonalenmasken*/
         rechts>>=1;
          if((*Testreihe&links)||/*ist besetzt?*/
             (*Testreihe&rechts)) return(besetzt);
       }
     return(frei);/* diagonalen sind frei*/
  }

main()
  { Setze_Dame(0,0);/* arbeiten tun immer*/
                    /* die anderen       */
    printf("es gibt %ld Loesungen\n",Anzahl);
  }
```

Lösungen (es kommt ja auf Zeit an). Damit ist wohl ein deutlicher Unterschied zu den KI-Sprachen zu sehen, ihre Werte lagen zwischen 13 und 130 Sekunden.

Das hergeleitete Programm nutzt dieselben Methoden wie die KI-Sprachen (Rekursion); der Geschwindigkeitsunterschied entsteht aus der optimalen Abstimmung der Operationen und Da tenstrukturen auf das Problem. Hierzu sind die KI-Sprachen nicht in der Lage, was aber nicht heißen soll, daß Probleme grundsätzlich besser auf herkömmliche Weise gelöst werden können.

Spielereien

Nachdem das Programm seinen Zweck erfüllt hat, können wir uns damit auseinandersetzen, dem Benutzer ein paar Bonbons für seine Aufmerksamkeit zu schenken. Als da wären: die Ausgabe der Lösungen oder sogar das bildliche Darstellen des "Damenwanderns".

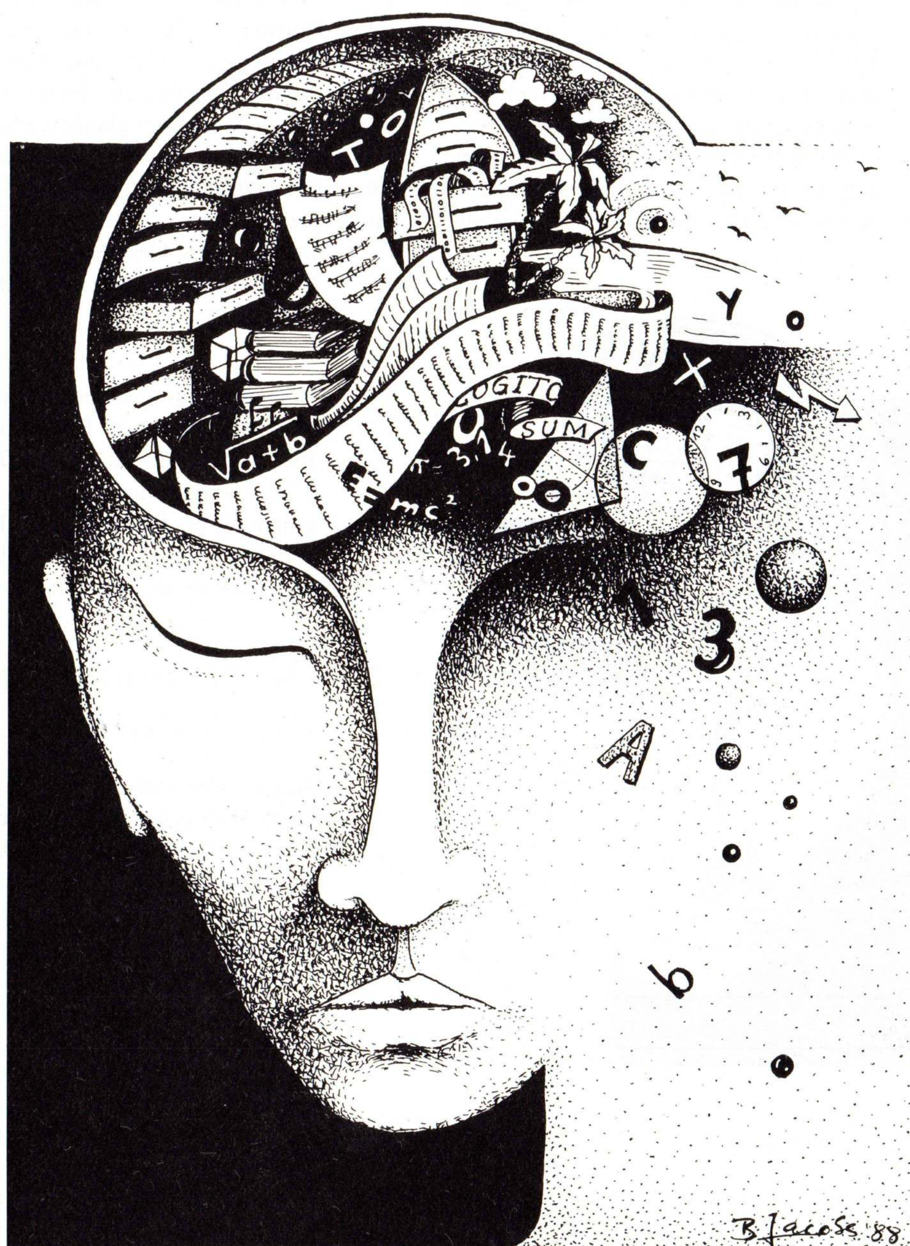
Beide Probleme sind sehr einfach lösbar, sofern der Leser rudimentäre Kenntnisse in 'C' besitzt und bereit ist, den beschriebenen Lösungsweg am Programm nachzuvollziehen. Hierbei ist zu beachten, daß das abgedruckte Listing eine erweiterte Version für verschiedene Feldgrößen ist.

Die Ausgabe einer Möglichkeit kann erfolgen, sobald wir eine Lösung gefunden haben; es wird, statt die Lösungsanzahl hochzuzählen, eine Ausgaberoutine aufgerufen (s. Listing 2). Diese holt das Zählen der Möglichkeiten nach und gibt danach die Lösung auf dem Bildschirm aus; hierbei wird auf einen Tastendruck gewartet, um ein Betrachten des Bildschirms zu ermöglichen.

```
#include <stdio.h>

#define MAX_FELD_GROESSE 10

int feldgroesse,spaltenmaske;
long anzahl;

setze_dame(feld,reihe,maske)
 register int *feld,reihe,maske;
 {int spalte,lokalfeld[MAX_FELD_GROESSE];
  register int k,links,rechts,*feldp,*lokalp;

  for(spalte=1;spalte<=spaltenmaske;spalte<<=1)
      { if((maske&spalte)==0)
          { feldp=feld+reihe;
            if((*feldp&spalte)==0)
               { if(reihe>=feldgroesse-1) anzahl++;
                 else
                   {lokalp=lokalfeld+reihe;
                    links=rechts=spalte;
                    for(k=reihe;k<feldgroesse-1;k++)
                     { links>>=1;
                       rechts<<=1;
                 *(++lokalp)=(*(++feldp)|links|rechts)&spaltenmaske;
                     }
                     if(lokalfeld[reihe+1]!=spaltenmaske)
                     setze_dame(lokalfeld,reihe+1, maske|spalte);
                   }
               }
          }
      }
   }

main()
{ int i,feld[MAX_FELD_GROESSE];

  spaltenmaske=0;
  for(feldgroesse=1;feldgroesse<=MAX_FELD_GROESSE;feldgroesse++)
      {for(i=0;i<feldgroesse;i++)feld[i]=0;
       anzahl=0L;
       spaltenmaske=(spaltenmaske<<1)+1;
       /*timerfunktion*/
       setze_dame(feld,0,0);

       printf("Feldgr%3d %6ld Loesungen\n",feldgroesse,anzahl);
       /*getchar();*/
      }
}
```

Die zweite Variante erfordert, daß jede Rekursionsebene ihre Reihe mit der entsprechend plazierten Dame selbst ausgibt. Die Ausgabe müßte also in der Schleife erfolgen, die die Dame über die Reihe schiebt. Wird die Rekursionsebene abgebrochen, so sollten die Felder der entsprechenden Reihe geleert werden, es gibt dann ja in dieser Reihe keine Dame mehr.

ENDE



DR. ADOLF EBELING

Von Menschen & Maschinen

Selten hat ein Forschungszweig der Wissenschaft so zwischenenthusiastischer Anerkennung seiner Befürworter und Anhänger einerseits und spöttisch-beißender, total ablehnender Kritik der Gegner andererseits gestanden wie die 'Künstliche Intelligenz' (KI), US-amerikanisch: 'Artificial Intelligence'. Dabei ist die Frage nach der bedingungslosen Schärfe der Kritik leicht zu beantworten. Bislang hatte jede Wissenschaftsrichtung die Möglichkeiten des Menschen erweitert, ihm ideelle und technische Werkzeuge an die Hand gegeben, mit der er die Welt besser erkennen und beherrschen konnte, nun aber schien mit der KI die Zeit angebrochen, in welcher das höchste Gut des Menschen, das sich ja in der Wissenschaft selbst verkörpert, nämlich der Intellekt, seine Einzigartigkeit und Spitzenstellung im Reich des Lebendigen verlöre, eben durch den Einsatz denkender Maschinen mitsamt ihren Programmen. Kurz: Der Mensch würde von wirklich intelligenten Maschinen überflügelt, die ihn dann, wie er es zuvor mit den restlichen Lebewesen auf der Erde getan hatte, auf eine tiefere Ebene evolutionärer Entwicklung zurückstuften.

Verständlich, daß in Kritik und Widerspruch tiefe Emotionen mitschwingen, schließlich geht es nicht um irgendeine neue technische Weiterentwicklung, sondern die Aussagen der KI betreffen jeden Kritiker persönlich, insbesondere dann, wenn er sich in seinem Menschsein an den bestehenden Maschinen oder Programmen messen lassen soll. Schon ein Taschenrechner ist ihm, bezogen auf die Rechenschnelligkeit, überlegen, aber kann man es einem Kritiker verübeln, jeden Vergleich mit einer einfachen Rechenmaschine als absurd abzutun? Und noch eine Demütigung muß der Kritiker ertragen: Sein freier Wille wird ihm abgesprochen. Die Entwicklung des Geistes über den Menschen hinaus zeigt überaus deutlich, so argumentieren die Befürworter der KI, daß der scheinbar freie Wille nur ein Hirngespinst des sich überhöhenden Menschenbewußtseins ist, in Wahrheit sind alle Abläufe in der Natur determiniert, nach Regeln, die man in den Naturgesetzen und der mathematischen Logik findet, auch das Chaos besitzt Systemeigenschaften. Die angeblich freien Entscheidungen erfolgen immer auf der Basis eines festgelegten Regelsatzes.

Worin nun aber liegt die Faszination der KI begründet? Nur im ersten Moment gut verborgen hinter einer oberflächlichen Kaltschnäuzigkeit, mit der KI-Apologeten ihre Forschungsergebnisse bis zum Zerreißen überstrapazieren, wenn sie Denken auf Gehirn und die Gehirnfunktionen auf einen Satz logischer Regeln reduzieren. Denn, steckt dahinter nicht das Verlangen derjenigen, die diese Regeln kennen und sie anwenden, auch das zukünftige Denken beeinflussen zu wollen? Ist hier nicht der uralte Wunschtraum nach einer vernunftgeregelten Welt zu entdecken und auch zu verstehen? Handelt man also durchaus im Einklang mit der Evolution, die ja im Sinne eines Fortschreitens zum Vollkommenen den Menschen und mit ihm eine höhere Form des Denkens und der Intelligenz hervorgebracht hat und nun über diese mangelhafte Ausstattung hinaus, zu einer noch höheren Stufe führen wird, zu einer emotionsfreien, kristallinen Maschinenintelligenz?

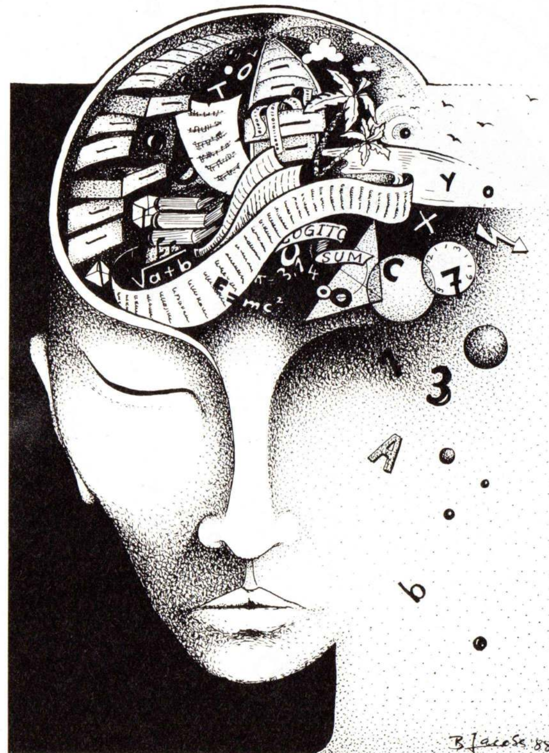
Doch der Reihe nach. Jede, auch die scheinbar originär in einem Geistesblitz geborene Idee oder Tat (ebenso als spontan, emergent, fulgurativ oder kreativ bezeichnet) bezieht ihre Grundlagen aus anderen vorausgegangenen Ideen oder Taten. Eine eindringliche Metapher stammt von Arthur Koestler, der eine Neuschöpfung mit "reculer pour mieux sauter" (zurückweichen, um besser springen zu können) umschreibt, d.h., um über ein vorliegendes Problem hinaus Neuland zu erreichen, muß man auf Bewährtes zurückgreifen, quasi Anlauf nehmen. Die KI hat Anlauf genommen aus der materialistischen Position im Meinungskampf des jahrhundertealten sogenannten 'Leib-Seele-Konflikts'. Hierin versammeln sich auf dem einen Pol die Verteidiger der idealistischen Position, die letztendlich an das Primat des Geistes glauben, an eine immaterielle Kraft, die allen biologischen Prozessen zugrunde liegt, während die Anhänger des anderen Extrems in letzter Instanz alles Geschehen auf die Gesetze der Materie zurückführen.

Von einem ihrer frühen Vertreter, dem französischen Philosophen Julien Offray de Lamettrie (1709-1751), wurde ein für diese Anschauung bis heute gültiges Bild geprägt: "L'homme machine" (der Mensch als Maschine).

Mechanistische Sprache

Was lag näher, als den umgekehrten Weg zu gehen, eben eine menschenähnliche Maschine zu bauen? Versuche gab und gibt es genug, man denke nur an die beweglichen Figuren von Vaucanson aus Grenoble, den Descartschen Automaten, den 'mechanischen Schachspieler' des Baron von Kempelen, den 'künstlichen Menschen' von Frederic Ireland und schließlich die 'intelligenten' Maschinen der KI. Gar nicht lange ist es her, daß die ersten Modelle der KI auf einfachen kybernetischen Regelkreisen beruhten. Man übersetzte Begriffe aus der Philosophie oder Psychologie direkt in die Sprache der Technik. Bewußtsein wurde so zum *'Adaptiven Monitoring'*, Denken zu *'Rechnen'*, Reflexion zu *'Rückkopplung'*, Begreifen zu *'Sortieren'* oder *'Adressieren'* usw. Ein anschauliches Beispiel liefert Salomón Klaczko-Ryndziun, indem er einen Text aus Kants *"Kritik der reinen Vernunft"* in die Sprache der Datenverarbeitung übersetzt:

Kant: *"Ohne eine zugrunde liegende Anschauung kann die Kategorie allein mir keinen Begriff von*



Wir könnten eine Maschine bauen, die alles tun wird, was Gehirne mit Informationen tun: Probleme lösen, Gefühle empfinden, Halluzinationen beim Fehlen sinnlicher Wahrnehmungen entwickeln, was sie nur wollen; vorausgesetzt, wir können eindeutig beschreiben, was das Gehirn tut.

einem Gegenstande verschaffen, denn nur durch Anschauung wird der Gegenstand gegeben, der hernach der Kategorie gemäß gedacht wird."

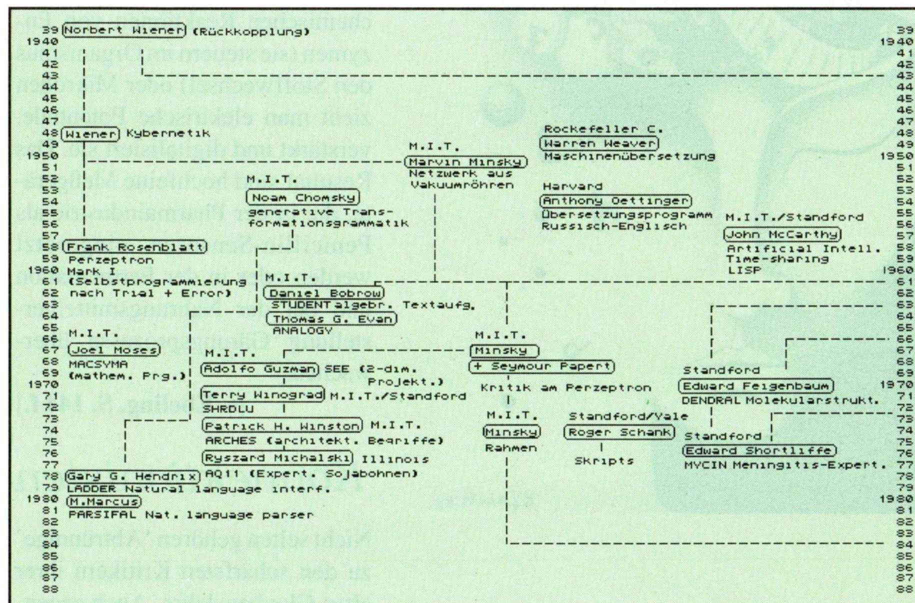
Klaczko-Ryndziun: *"Ohne eine zugrunde liegende Eingangsinformation können die Algorithmen allein mir keine Blockadresse zum Klassifizieren des zugehörigen Gegenstandes verschaffen; denn nur durch die Eingangsinformation wird der Gegenstand beschrieben, der hernach dem Algorithmus gemäß beschrieben wird."*

[Klaczko-Ryndziun, S. 13 f.]

ich" [**Hofstadter, S. 363**]; und der Computerwissenschaftler Martin H. Schulz fordert gar die Abschaffung der experimentellen Wissenschaften zugunsten von Computersimulationen.

Alles-oder-Nichts-Prinzip und Von-Neumann-Rechner

Denken als Folge von elementaren Summierungen, neurologisch und mathematisch beschreibbar, diese Vorstellung hat



Diese Art, Erkenntnisprobleme mit einer angeblich vom Nebel des Idealismus gereinigten, materialistisch klaren Sprache zu beschreiben und mathematisch-technisch zu lösen, durchzieht die gesamte Entwicklung der KI, von ihren geschichtlichen Vorläufern bis zu den gegenwärtigen Varianten. Schon George Boole erhob seine algebraische Logik zu *"Gesetzen des Denkens"* (*"Laws of Thought"* lautet der Titel seines mathematisch-logischen Hauptwerkes); Wilhelm Gottfried Leibniz wollte die Verständnisschwierigkeiten zwischen Wissenschaftlern durch die Sprache der Mathematik lösen; im Einklang mit dem Logiker Alonzo Church hielt Alan M. Turing alles, was berechenbar ist, auch für berechenbar mit Hilfe seiner nach ihm benannten Maschine, dem Urmodell aller Computer; Douglas R. Hofstadter wandelt halb im Scherz, halb im Ernst das Descartsche Credo *"Cogito, ergo sum"* (Ich denke, also bin ich) um in *"Ich denke, also sum(miere)*

die KI bis in die letzten Jahre hinein geprägt. Ausgangspunkt waren die Forschungen von Warren S. McCulloch und Walter H. Pitts. In ihrem Neuronenmodell wird Information in einem Alles-Oder-Nichts-Prinzip weitergeleitet, entweder eine Nervenzelle feuert oder sie schweigt. Neben Nervenerregungen entdeckte man auch Hemmprozesse, beide zusammen, summiert zu unzähligen Nervenimpulsen, hieß es, bilden jede Information ab, auch hochkomplexe. Wenn man also die Grundbedingungen neuraler Informationsübertragung erkannt hat, dann lassen sich Gehirnfunktionen, damit Denkvorgänge und Bewußtseinsprozesse etc. beschreiben und in die Sprache einer Maschine übersetzen. Mit McCullochs Worten:

"Wie von Neumann zeigte, könnten wir eine Maschine bauen, die mit Informationen alles tun wird, was Gehirne mit Informationen tun: Probleme lösen, Gefühle empfinden, Halluzinationen beim Fehlen sinnlicher Wahrnehmungen entwickeln, was sie nur wollen; vorausgesetzt, wir können in einer endlichen und eindeutigen Form das beschreiben, was (das Gehirn) unserer Meinung nach tut."

[Klaczko-Ryndziun, S. 276]

Endlich und eindeutig muß die Beschreibung sein, und, was implizit mitschwingt, sie muß in sich einen Algorithmus bergen, einen Ablaufplan, der das Funktionswerk der zu konstruierenden Maschine der Vorgabe entsprechend steuern kann. Nur, was in aller Welt ist eindeutig und endlich? Nicht einmal die Grundlagen der Mathematik, der Veste aller Wissenschaften, sind prinzipiell endlich erklärbar (wie uns der Mathematiker Kurt Gödel eindrücklich gezeigt hat), allenfalls axiomatisch festzulegen und zu begrenzen, und das bislang Beständigste, das Ganze, unser Universum entpuppt sich schon als nebensächliche Abteilung innerhalb eines unübersehbaren Polyversums.

Bis zum Aufkommen der Parallelrechner beruhten informationsverarbeitende Maschinen, von einigen Ausnahmen abgesehen, auf dem Von-Neumann-Prinzip (benannt nach dem Mathematiker John von Neumann). Eine solche Rechnerarchitektur setzt logisch und räumlich zerlegbare Funktionseinheiten wie einen Prozessor, Speicher, Ein- und Ausgabeeinheiten etc. voraus. Erst ein Programm, bestehend aus einer Reihe aufeinanderfolgender Befehle, abgelegt in Speicherzellen, macht die Maschine arbeitsfähig. Der Prozessor, die Zentraleinheit des Rechners, hat die Anweisungen entsprechend dem im Programm festgelegten Ablaufplan Schritt für Schritt auszuführen.

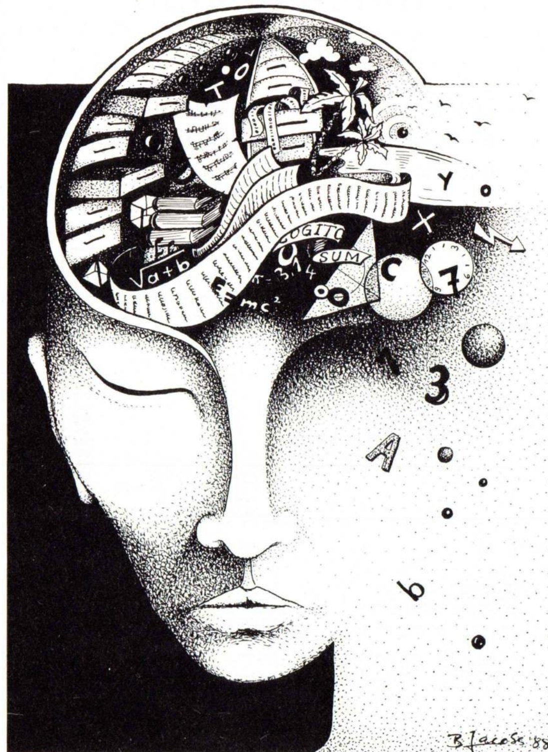
Natürlich leistet eine derartig konstruierte Maschine Rechenarbeit, die bis dato nur dem menschlichen Intellekt zugetraut wurde, aber daraus zu schließen, unser Gehirn müsse ähnlich funktionieren, das grenzt entweder an Einfalt oder an gewollte Ignoranz. Wo findet man im Gehirn die scharfe Trennung zwischen Prozessor und Speicher, wo überhaupt Speicherzellen? Das ganze Gehirn, und das wissen wir aus zahlreichen Untersuchungen, ist Gedächtnis. Auf keiner Stufe stimmt der Vergleich zwischen Maschine und Gehirn, weder bei den Basisstoffen (Silizium versus Kohlenstoff) noch bei den Arbeitseinheiten noch bei der Pro-

grammlogik insgesamt. Warum sollten im Gehirn Informationen binär kodiert werden, nur weil man ein Alles-Oder-Nichts-Prinzip postuliert hatte? Ein heute längst überholtes Postulat: Nervenimpulse pflanzen sich nicht allein digital fort, allerorten findet man fließende Übergänge zwischen diskret und stetig. Nervennetze arbeiten sowohl sequentiell als auch parallel, sowohl digital als auch analog, sowohl zeitlich als auch räumlich, sowohl elektronisch als auch chemisch.

Biocomputer

Wie gerufen zeichnet sich am Forschungshorizont schemenhaft das Gespenst einer Verschmelzung künstlicher Schaltung mit Biomaterie ab. Silizium, auf der Erde ausreichend vorhanden, läßt sich 1. in solch hoher Präzision herstellen, daß nicht mehr als ein oder zwei fremde Atome in einer Milliarde Siliziumatomen zu finden sind und 2. in ein beinahe ideales Kristallgitter rastern und erlaubt damit, eine riesige Zahl von Funktionselementen auf kleinstem Raum zu speichern. Ein *VLSI-Chip* (Very Large ScaleIntegration) vereint auf einer Fläche von kaum einem Quadratzentimeter eine Million Schaltelemente. Beim Fortdauern der Geschwindigkeit zunehmender Miniaturisierung jedoch würden in absehbarer Zeit physikalische Grenzen erreicht; um sie zu überschreiten, wäre ein immenser technischer Aufwand nötig, der in keinem Verhältnis zum Ergebnis stehen würde. Das Interesse zukünftiger Forschung könnte sich deshalb dem Gebiet der Bio-Elektronik zuwenden. Geeignete Biomoleküle erreichten bei möglicher Realisierung durch tausendfach kleinere Strukturen und Ausnutzung von dreidimensionaler Anordnung der Moleküle eine Leistung von Millionen oder mehr Chips gleicher Größe. Der englische Begriff 'Molecular Electronic Devices' faßt präziser, um was es geht: nicht um die Produktion lebender (Bio-) Chips, sondern um die Nutzung organischen Materials als binäre Schalter bzw. Speicherelemente.

Bestimmte organische Moleküle, Eiweißketten oder Enzyme, können zwei elektrisch unterscheidbare Zustände annehmen, somit einen binären Wert abbilden. Aufgrund dessen sehen, noch phantastische, tief in die Zukunft gerichtete Ideen vor, Proteine als Schaltelemente auf binärer Basis zu konstruieren und als Bauplan eines Biocomputers in die DNS



Ohne eine zugrunde liegende Anschauung (Eingangsinformation) kann die Kategorie (Algorithmus) allein mir keinen Begriff (Blockadresse zum Klassifizieren) von einem Gegenstande verschaffen, denn nur durch die Anschauung (Eingangsinformation) wird der Gegenstand gegeben, der hernach der Kategorie (dem Algorithmus) gemäß gedacht wird.

(Desoxyribonukleinsäure, der stoffliche Träger der Erbinformation) von Bakterien einzubauen. Ein solch geschaffener Kern eines Biocomputers könnte sich nach den eingepflanzten Plänen selbst konstruieren. Den technischen Schwierigkeiten zur Realisierung dieses Unternehmens gesellen sich auch zukünftig noch weitgehende Unkenntnis über die semantischen (die Bedeutung betreffenden) Funktionen des genetischen Kodes zu, abgesehen von den ethisch-moralischen Fragen, die an erster Stelle beantwortet werden müssen.

Anwendungsreif ist die Forschung im Bereich der Biosensoren. Aus chemischen Reaktionen von Enzymen (sie steuern im Organismus den Stoffwechsel) oder Mikroben zieht man elektrische Potentiale, verstärkt und digitalisiert sie. Das Resultat sind hochfeine Meßgeräte, die in der Pharmaindustrie als Penicillin-Sensoren eingesetzt werden oder in der Fermentation und bei der Nahrungsmittelherstellung Gärungsprozesse überwachen.

[s. Ebeling, S. 144 f.]

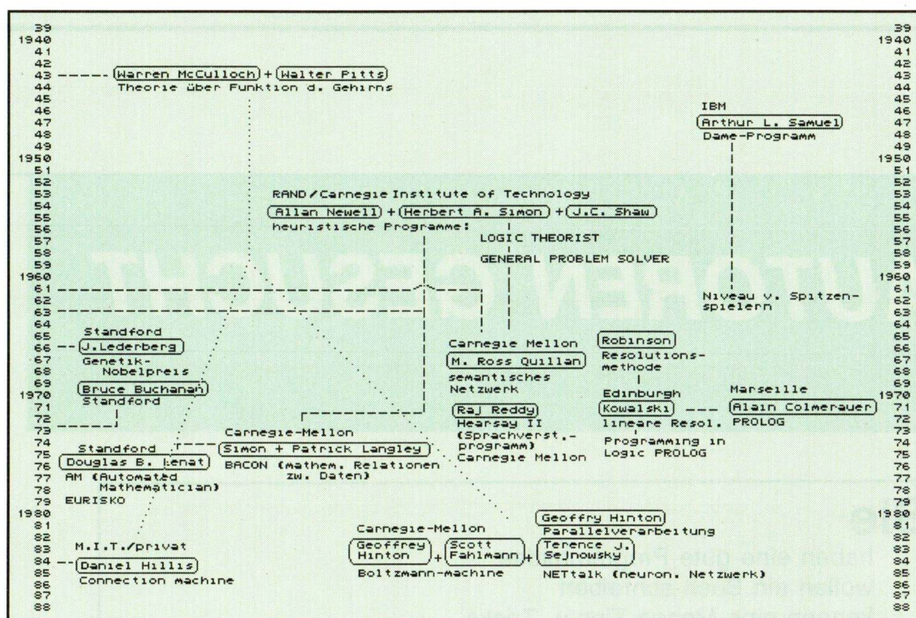
Konnektionisten

Nicht selten gehören 'Abtrünnige' zu den schärfsten Kritikern ihrer alten Glaubenslehre. Auch gegenwärtig kommt massive Kritik an der klassischen KI von Wissenschaftlern, die sich von ihr ab- und den sogenannten *'Neuro-Computern'* zugewandt haben. Konnte in der Frühzeit der KI einer ihrer Begründer, John McCarthy, bereits einem Einfachst-Automaten, wie zum Beispiel einem Thermostaten, Überzeugungen zubilligen oder noch zu Beginn der 80er Jahre Geoff Simons vom digitalen, bedingte Sprünge ausführenden Computer als einem Lebewesen mit selbständigen Willensentscheidungen sprechen, so hört man heute aus den Reihen der KI-Avantgarde ('Konnektionisten' wie Daniel W. Hillis, Terrence J. Sejnowsky, James L. McClelland u. a.), natürlich funktioniere ein Gehirn nicht wie ein sequentieller Computer, nahezu alles an den herkömmlichen Vergleichen sei falsch.

Man müsse endlich der Natur folgen, heißt es, ihre Erfolgsprinzipien übernehmen, und jene seien nun mal durch parallel laufende Prozesse gekennzeichnet. Vergleichen wir: Auch der leistungsfähigste, auf sequentieller Basis gebaute Superrechner braucht noch mehrere Jahrzehnte, wenn nicht gar ein ganzes Jahrhundert, um die normale Funktion der menschlichen Retina nachzuahmen, die diese in einer einzigen Sekunde ausführt. Zwangsläufig also müsse der Parallelverarbeitung die Zukunft gehören. Erste Schritte sind bereits getan, angefangen von Vektorrechnern über Transputersysteme bis hin zu Daniel Hillis' *'Connection Machine'*, einer Zusammenschaltung von rund 65.000 Prozessoren, jeder mit seinem eigenen Speicher ausgerüstet.

Connection Machine

Was ist neu an den neuen Konzepten? Zumindest die Hardware-Ausstattung. Nehmen wir als Beispiel die Connection Machine. Sie besteht aus acht miteinander verbundenen, würfelförmigen Kästen mit einer Gesamtlänge von 1,5 Metern. Jeder Würfel enthält 16 vertikal angeordnete Karten, auf denen sich 32 Chips befinden. Jeder Chip wird auf einer Seite des Würfels durch einen Lichtpunkt gekennzeichnet, über den sein Status abzulesen ist. In einem Chip sind 16 Prozessoren eingearbeitet mit je einer kleinen Speichereinheit von 512 Bytes. Insgesamt bilden also 65.536 Prozessoren die Basis der Connection Machine. Alle Prozessoren miteinander zu verknüpfen, ist praktisch unmöglich. Nach dem Modell eines 12-dimensionalen Würfels mit 212 (4096) Ecken hat man die Einheiten so zusammengeschaltet, daß in höchstens 12 Schritten eine Botschaft (zwischen zwei beliebigen Ecken) ihr Ziel erreichen kann. Sollte eine Route gerade belegt sein, hält der Rechner ohne Verzögerung eine Ausweichstrecke parat. Die Connection Machine zählt zu den schnellsten Rechnern der Welt, mehrere Milliarden Instruktionen pro Sekunde kann sie bearbeiten.



Zur zukunftsweisenden Hardware gehört eine ebensolche Software. Im konnektionistischen Zweig der KI glaubt man seit längerem nicht mehr an die alleinige Erkenntniskraft algorithmischer Verfahren, ihrer Grenzen wurde man allenthalben gewahr. Zu neuen Ufern, in noch unerschlossene Bereiche künstlicher Intelligenz will man nun mit Hilfe von der Natur nachempfundenen, sich selbst organisierenden Systemen gelangen. Fieberhaft wird in der Echtzeit-Bildverarbeitung und der Spracherkennung mit den neuen Modellen gearbeitet.

Assoziationen

Assoziation heißt das Zauberwort. Spätestens seit den Experimenten des Neurologen Karl Pribram, der zeigen konnte, daß Gedächtnisinhalte aus den verschiedensten Teilen des Gehirns reproduzierbar sind, weiß man, daß Informationen im Gehirn nicht in einzelnen Symbolen abgespeichert werden. Das Symbol 'Großmutter' wird eben nicht in irgendeiner Nervenzelle abgelagert, aus der man es gezielt jederzeit zurückholen kann; im Gegenteil, tausendfache multimedial miteinander verknüpfte Eindrücke, Bilder, Gefühle, Worte in einer sich ständig variierenden Umgebung formen den Eindruck eines Symbols. Und schon wenige Anteile dieses Assoziationskomplexes können lange Erinnerungssequenzen abrufen. Ein jeder kennt die Erfahrung, wenn ein längst vergessen geglaubter Duft nach vielen Jahren in einer gänzlich anderen Situation als die erinnerte, unversehens uralte Bilder aus den Tiefen des Gedächtnisses hervorholt.

Warum sollten dann nicht per Software erzeugte assoziative Felder, in denen man Daten so miteinander verknüpft, daß bei der Suche bereits Teilinformationen zum Gesamtbild führen, ähnliche Funktion erfüllen wie das natürliche Gedächtnis? Ist doch der Grad des Unterschiedes vermeintlich nur ein quantitativer, eine Diskrepanz, die mit fortschreitender Verbesserung der Rechenleistung und Einbeziehung immer zahlreicherer Computer in das Rechennetzwerk irgendwann aufgehoben werden kann und die sich danach sogar zugunsten der Maschinen verschieben wird. Die mathematischen Formeln jedenfalls kennt man bereits.

Das erinnert an die Raumfahrt. Seit Jahrzehnten schon füllen ab und an schön gezeichnete Bilder von riesigen Städten im Weltraum die Magazinseiten, werden bemannte Raumfahrtprojekte vorgestellt, die tief hinein in unbekannte Fernen des Alls führen sollen, doch in der Realität hat es nicht weiter als bis zum Mond gereicht. Und bei der KI? David Levy, selbst ein exzellenter Schachcomputerentwickler, hat seine berühmte Wette mangels Gegner eingestellt, nämlich mehr als 100.000 Dollar demjenigen zu zahlen, dessen Schachcomputer den amtierenden Weltmeister schlägt. Hatte man nicht in der Anfangszeit der KI prophezeit, bald unschlagbare Schachautomaten zu bauen, sprachverständige Maschinen zu konstruieren, mit denen automatische Sprachübersetzung möglich würde, neue Gesetze der Mathematik durch logikgestützte Programme zu entdecken, die Psychologie auf die Theorie von Computerprogrammen zurückzuführen und die internationale Politik mittels heuristischer Verfahren zu regeln?

Erkenntnis

Verfrühte Euphorie, sagt man heute und betont im gleichen Atemzug, daß mit dem Einsatz sogenannter Neuro-Computer eine umwälzende Entwicklung zur Erklä-

rung des Gehirns einsetzen werde. "Der Computer erkennt mein Gesicht", begeistert sich der Synergetiker Hermann Haken.

[**Bild der Wissenschaft, 1988/8,**]

Natürlich erkennt der Computer kein Gesicht, vielmehr rastert ein Programm eine Photovorlage ab und kann diese aufgrund von Teilstücken reproduzieren. Ein Computerprogramm erkennt nichts, genauso wenig wie ein Schachprogramm etwas vom Schachspiel versteht oder die Regeln kennt. Der Computer versteht das, schreibt der Linguistiker John Searle, *"was das Auto und die Rechenmaschine verstehen, nämlich haargenau nichts. Das Verstehen des Computers ist nicht bloß bruchstückhaft oder unvollständig; es ist gleich null."*.

[**Hofstadter u. Dennett, S. 342**]

Im gleichen Sinne kann die Frage, ob die starr verschalteten Kleinstrechner von Hillis mit einer in den Anfängen steckenden Software ein angemessenes Modell des Gehirns und des neuralen Geschehens - einer *"dynamischen Komplexität, die unermeßlich größer ist als irgendetwas, das jemals im Universum entdeckt worden oder in der Computertechnologie geschaffen worden ist"* [**Sir John C. Eccles, in: Popper und Eccles, S. 301**] - liefern, nach bisherigen Wissen über das Gehirn mit Entschiedenheit verneint werden. Ein winziges im Vergleich zur Komplexität des Gehirns geradezu unbedeutendes Virus namens AIDS bereitet Spitzenforschern schon seit Jahren bislang unüberwinbare Schwierigkeiten, aber die Funktionen des Gehirns wird man bald im Griff haben. Gemessen an ihren Ansprüchen und dem Aufwand, mit dem geforscht wurde, hat die KI seit ihrem Bestehen so gut wie nichts erreicht. Die wenigen brauchbaren Expertensysteme oder die Maschinenübersetzung, die allesamt der überprüfenden Hand des Menschen bedürfen, hätte man sicher auch ohne KI programmieren können. Wie lange also wollen wir noch auf die künstliche Intelligenz warten?

Literatur:

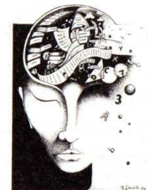
Ebeling, A., Gehirn, Sprache und Computer, Heise, 1988.

Hofstadter, D. R., Gödel, Escher, Bach, Klett-Cotta, 1985.

Hofstadter, D. R.; Dennett, D. C., Einsicht ins Ich, Klett-Cotta, 1986.

Klaczko-Ryndziun, S., Systemanalyse der Selbstreflexion, Birkhäuser, 1975.

Popper, K. R.; Eccles, J. C., Das Ich und sein Gehirn, Piper, 1985.



ABO

ABO

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim Verlag
Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57

Einzelheft- u. Monatsdisketten Bestellung

Einzelheft- u.
Disketten Service

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte
mit
60 Pf.
frankieren

Heim Verlag
Heidelberger Landstr. 194

6100 Darmstadt-Eberstadt
Telefon 0 61 51 / 5 60 57

Kontaktkarte

Kontaktkarte

Bitte Adresse der Firma, bei der Sie Informationen, oder etwas bestellen möchten, auf der rechten Seite eintragen → → → → → → → → → →

Absender
(Bitte deutlich schreiben)

Vorname/Name

Beruf

Straße/Nr.

PLZ/Ort

Telefon Vorwahl/Rufnummer

Postkarte

Bitte
freimachen

Firma

Straße/Postfach

PLZ Ort

KICK START AMIGA ZEITSCHRIFT

Abonnement

Ja, bitte senden Sie mir die Amiga-Computer Fachzeitschrift ab ______
für mindestens 1 Jahr (11 Hefte) zum ermäßigsten Preis von jährlich DM 70,— frei Haus.
(Ausland: Nur gegen Scheck-Voreinsendung DM 90,— Normalpost.)
Der Bezugszeitraum verlängert sich nur dann um ein Jahr, wenn nicht 6 Wochen vor Ablauf des Abonnements gekündigt wird.

Name ______

Vorname ______

Straße/Nr. ______

PLZ ______ Ort ______

Gewünschte Zahlungsweise bitte ankreuzen

☐ Bequem und bargeldlos durch Bankeinzug

Konto-Nr. ______ BLZ ______

Institut ______ Ort ______

☐ Ein Verrechnungsscheck über DM ______ liegt bei.

Garantie:
Diese Bestellung kann ich schriftlich innerhalb einer Woche (rechtzeitige Absendung genügt) widerrufen. Dies bestätige ich durch meine 2. Unterschrift.

Datum ______ Unterschrift ______

Datum ______ Unterschrift ______

KICK START AMIGA ZEITSCHRIFT

ABO

KICK START AMIGA ZEITSCHRIFT

Einzelheft- u. Monatsdisketten Bestellung

KICKSTART können Sie direkt beim HEIM-VERLAG zum Einzelheft-Preis von DM 7,– (zuzüglich Gebühr für Porto und Verpackung) nachbestellen. Bearbeitung nur gegen beigefügten Scheck über den entsprechenden Betrag (keine Überweisung).

Jan.	Febr.	März	April	Mai	Juni	Juli	Aug.	Sept.	Okt.	Nov.	Dez.	
												1987 = DM
												1988 = DM
												1989 = DM

+ Gebühr für Porto u. Verp. ______

☐ Scheck in Höhe ______ zus. DM ______ liegt bei

Disketten Service

Alle Programme, die in KICKSTART veröffentlicht wurden, sind auf Disketten erhältlich. Die Disketten enthalten die Programme von jeweils 2 KICKSTART-Ausgaben. Bestellen Sie durch ankreuzen die gewünschten Disketten

Preis je Diskette 19,– DM	Juli/Aug.	Sept./Okt.	Nov./Dez.	Jan./Febr.	März/April	Mai/Juni	Juli/Aug.	Sept./Okt.
	87	87	87	88	88	88	88	88
	Nov./Dez.	Jan./Febr.	März/April	Mai/Juni	Juli/Aug.	Sept./Okt.	Nov./Dez.	Jan./Febr.
	88	89	89	89	89	89	89	90

Lieferung: gegen beigefügten Scheck zuzügl. 5,– DM Versandkosten (unabhängig von bestellter Stückzahl)

KICK START AMIGA ZEITSCHRIFT

Einzelheft- u. Monatsdisketten Bestellung

KICK START AMIGA ZEITSCHRIFT

☐ Ich bitte um weitere Informationen

☐ Ich gebe folgende Bestellung auf

in Bezug auf Ihre Anzeige in Kickstart Heft ______ Seite ______

Menge	Produkt/Bestellnummer	DM	gesamt DM

Datum, Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

Abgesandt am:

KICK START AMIGA ZEITSCHRIFT

Firma: ______

Bemerkungen: ______

KICK START

AMIGA ZEITSCHRIFT

'TOP 12'

Mein Lieblingsspiel

Meine Meinung

Zu dem Artikel ____________ in Heft ___, Seite ___
hätte ich folgendes zu bemerken:

☐ Ich möchte Ihnen folgendes Programm zur Veröffentlichung anbieten: (Kurzbeschreibung, Sprache, Länge in Druckerseiten)

☐ Ich kann über folgendes Thema berichten: (Tips & Tricks am AMIGA, Hardware, Software, etc.)

☐ Ich möchte gerne Autor in der KICKSTART werden. Meine Fachgebiete: (z.B. LISP, Modula-2, Assembler...)

☐ Ich möchte, daß folgendes Public-Domain Programm in Ihre Sammlung aufgenommen wird.

☐ Sonstiges

Bei weiteren Angaben oder Fragen wenden Sie sich bitte schriftlich oder telefonisch an die Redaktion. Tel. 0 61 96/48 18 11

Kurzmitteilung

PUBLIC DOMAIN SERVICE

Ich bestelle folgende PD-Disketten:
(Siehe PD Service in dieser Ausgabe)

Zahlung erfolgt:

☐ per Scheck

☐ per Nachnahme

(nur Inland) zuzüglich DM 3,70 Nachnahmegebühr

Ab 5 Disketten Versandkosten frei

Je Diskette fügen Sie bitte einen Betrag von DM 8,– bei,
für Porto und Verpackung je Sendung DM 5,– (Ausland DM 10,–)

Datum Unterschrift

PD Bestellung

Datum

'TOP 12'
Mein Lieblingsspiel

'TOP 12'
Mein Lieblingsspiel

Wettbewerbsbedingungen siehe TOP 12 in diesem Heft

Bitte freimachen

MAXON - Computer GmbH
Redaktion KICKSTART
'TOP 12'
Industriestraße 26

D-6236 Eschborn

Kurzmitteilung

Kurzmitteilung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Telefon

Postkarte

Bitte freimachen

MAXON Computer GmbH
KICKSTART Redaktion
Industriestraße 26

6236 Eschborn

PD Bestellung

PD Bestellung

Absender
(Bitte deutlich schreiben)

Vorname/Name

Straße/Nr.

PLZ/Ort

Postkarte

Bitte freimachen

MAXON Computer GmbH
KICKSTART Redaktion
Industriestraße 26

6236 Eschborn

KICKS FÜR INSIDER

Kaum ist die närrische Zeit vorbei, da treiben die AMIGA-Programmierer wieder ihr Unwesen. Doch keine Angst, die Programme entstanden weit vor Rosenmontag. Damals dachte noch keiner daran, daß während dieser Zeit manch einer auch ohne unser interessantes Copper-Listing die Workbench mit 2048 Farben sehen konnte.

Unser AMIGA-Freak behält jedenfalls einen klaren Kopf und jongliert vornehmlich mit Bits und Zeigern, wobei er ebensoviel Fingerspitzengefühl und Kopfarbeit leisten muß, wie bei der abgebildeten Tätigkeit.

INHALT

VON ROGER FISCHLIN

Textscrolling

Schnelles Scrollen von flotten Texten

64er-Veteranen dürften die Scroll-Texte wohl bekannt sein. Auch auf dem AMIGA sind sie in steigender Zahl zu sehen. Doch viele Programmierer schrecken vor Scroll-Texten zurück, da sie befürchten, deren Programmierung könne beim AMIGA im Gegensatz zum 64er schwierig sein. Daß dies jedoch nicht der Fall ist, beweist das rund 125 Zeilen lange Assemblerprogramm.

Müssen Blitter und Copper in Anspruch genommen werden?

Der Einsatz des Blitters zum Kopieren der Bits und des Coppers zum teilweisen Bildschirmscrollen sind nicht nötig. Die Bits werden mit dem ROXL-Befehl verschoben, und weil dies bitweise geschieht, muß auch nicht ein Teil des Bildschirms mit Hilfe der Hardware in Bit-Schritten bewegt werden. Der ROXL-Befehl verschiebt die Bits nach links, das Systembit X wird nachgeschoben, das herausgeschobene Bit geht u.a. ins Systembit X. Leider läßt der Assembler den Befehl nur als Word-Befehl gelten.

Ein Beispiel:

```
Register   d0 : $1
           d1 : %0100110011010101
Systembit   X : 1

Befehl        : ROXL.W d0,d1

Register   d0 : $1
           d1 : %1001100110101011
Systembit   X : 0
```

Um nun eine Zeile um ein Bit nach links zu verschieben, muß mit dem ROXL-Befehl jedes Word vom letzten bis zum ersten Word der Zeile um ein Bit verschoben werden. Durch diese Reihenfolge gelangt das 15. Bit des vorherigen Words ins X-Systembit und dadurch beim folgenden Word ins Bit 0. Rechts muß nun ein neues Bit des nächsten Buchstabens erscheinen, während das links herausgeschoben Bit wegfällt.

Wie gelangt man an die Zeichensatzdaten?

Das Longword nach dem 34. Byte des Zeigers, den man beim Öffnen des Fonts erhält, zeigt auf die Daten. Die Daten sind aber zeilenweise abgelegt, d.h. zunächst kommen die obersten Bytes jedes Buchstabens, dann die zweiten Bytes etc.. Die Daten eines Zeichens folgen also in einem festen Abstand. Er ist im Word nach dem 38. Byte definiert. Bei dem Systemfont TOPAZ-80 beträgt dieser $c0, da die Zeichen mit den ASCII-Code 0-$1f und $80-$9f nicht definiert sind. Da TOPAZ-80 8 Bit (=1 Byte) breit ist, entfallen Bitmanipulationen, um die eigentlichen Bits des Zeichens herauszufiltern.

Bei einem mit diesem Wissen programmierten Scroll-Text würde das ruckhafte Scrollen negativ auffallen. Der Grund dafür ist einfach: Während DMA das Bild auf den Monitor bringt, wird der Text verschoben. Die Folge ist, daß ein Teil des Textes verschoben und ein anderer Teil nicht verschoben dargestellt wird. Dies kann umgangen werden, indem man nur dann den Text verschiebt, wenn das DMA ihn bereits dargestellt hat. Die aktuelle Zeile, die das DMA gerade ausliest, ist in VHPOS (= $dff006) und VPOS (= $dff004) zu finden:

VHPOS Bit Nr. :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Funktion :	v7	v6	v5	v4	v3	v2	v1	v0							
VPOS Bit Nr. :	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
Funktion :															v8

Die Bits 0-7 sind also in VHOPS und das 8. Bit in VPOS zu finden.

Zum Programm

Das Programm entstand mit dem Devpac-Assembler. Eventuell muß der Pfadname der Include-Files geändert werden (z.B. in df0:). Zunächst werden die Intuitions- und Graphics-Library geöffnet. Da diese Libraries resident sind, sind sie immer vorhanden. Anschießend werden ein Screen, ein Window und der TOPAZ-80-Font geöffnet. A5 zeigt auf das Zeichen, das am rechten Rand hereingeschoben werden soll. Wieviele Bits bereits von diesem Zeichen hereingeschoben wurden, ist in d5 zu finden.

```
        lea.l     TEXT,a5
        moveq.l   #0,d5
```

Ist der Wert in d5 = 8, d.h. das Zeichen (Breite 8) ist vollständig zu sehen, so muß ein neuer Buchstabe erscheinen.

```
R1      and.l     #$7,d5
        bne       R4
```

A2 zeigt auf die untersten (7.) Bytes der Zeichendaten.

```
        move.l    FONT,a2
        move.l    34(a2),a2
        add.l     #7*$c0,a2
```

Ist das Textende (0 Byte) erreicht, soll wieder von vorne begonnen werden.

```
        moveq.l   #0,d0
R6      move.b    (a5)+,d0
        bne       R5
        lea.l     TEXT,a5
        bra       R6
```

Da die Zeichen mit den ASCII-Codes von 0 bis $1f nicht definiert sind, ist das Zeichen mit dem ASCII-Code $20 jeweils am Anfang der Datenzeilen zu finden. Auch die Zeichen mit den ASCII-Codes von $80 bis $9f sind nicht definiert, so daß man diese Zeichen ebenfalls in den Datenzeilen überspringen muß.

```
R5      cmp.b     #$7f,d0
        bls       R9
        sub.b     #$20,d0
R9      sub.b     #$20,d0
        add.l     d0,a2
```

Der Scroll-Wert muß um 8 erhöht werden, da ROXl nur mit Word arbeitet, der Wert aber nur als Byte vorliegt. A0 zeigt in der Bitplane auf die achte Zeile.

```
R4      or.b      #$8,d5
        move.l    Screen,a1
        move.l    $c0(a1),a0
        add.l     #8*40,a0
```

Es sind achtmal (Zähler in d1) jeweils eine Zeile mit 20 Words zu verschieben.

```
        move.l    #7,d1
        move.l    a2,a1
```

Es muß vorher aber auf eine geeignete DMA-Position gewartet werden. Das 9. Bit ist dabei nicht wichtig.

```
R7      move.w    $dff006,d0
        lsr.w     #8,d0
        cmp.b     #$50,d0
        bls.s     R7
```

Pro Zeile sind 20 Words zu verschieben (Zähler in d0). Das Byte des erscheinenden Buchstabens wird nach D2 gebracht und so oft nach links verschoben, bis das richtige Bit im Systembit X ist. Nun muß nur noch A1 auf das vorhergehende Zeichensatzbyte zeigen.

```
R2      move.l    #19,d0
        moveq     #0,d2
        move.b    (a1),d2
        roxl.w    #1,d2
        roxl.w    d5,d2
        sub.l     #$c0,a1
R8      roxl.w    -(a0)
        dbra      d0,R8
        dbra      d1,R2
```

Nach einer kurzen Verzögerung verschiebt das Programm den Text wieder um ein Bit weiter, es sei denn, der Benutzer hat die Maustaste gedrückt.

```
        move.l    #$e00,d0
R3      dbra      d0,R3
        addq      #1,d5
        btst      #6,$bfe001
        bne       R1
```

Es werden jetzt nur noch das Window und der Screen geschlossen.

Anpassung an das eigene Progamm

Soll der Scroll-Text sich schneller über den Bildschirm bewegen, muß nur der Wert ($e00) bei der Verzögerung verringert werden. Soll der Text an einer anderen Stelle auf dem Bildschirm erscheinen, ist der Zeiger auf die Stelle in der Bitplane (siehe oben!) zu ändern.

Es gilt:

```
add.l      #(8+Zeile)*40,a0
```

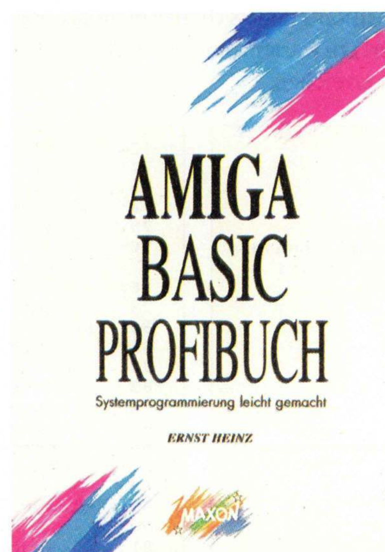
Es ist außerdem nötig, auf die DMA-Position Zeile+8 (lieber mehr als weniger!) zu warten, um ein weiches Scrollen zu erreichen.

Der Text kann ebenfalls geändert werden, er muß nur mit einen 0-Byte schließen. Da das Demo nur eine Bitplane verschiebt, sollte man bei mehreren Bitplanes als Hintergrund der Schrift die Farbe 0 und als Zeichenfarbe die Farbe 1 verwenden. Damit ist das Problem, welches schwierig aussah, doch auf einfache Weise zu lösen gewesen ...

```
* Scroll-Text
* Autor Roger Fischlin
*(c) Maxon Computer GmbH 1989
* Kickstart 1989

    incdir  "vd0:include/"    ;muß unter Umständen
                               geändert werden
    include exec/exec_lib.i
    include intuition/intuition.i
    include intuition/intuition_lib.i
    include graphics/graphics_lib.i
    include libraries/dos.i
    include libraries/dos_lib.i
    include graphics/text.i

    lea     int_name(pc),a1 ;
Intuition-Library
    CALLEXEC     OpenLibrary ; öffnen
    tst.l   d0
    beq     quit

    move.l  d0,_IntuitionBase
    moveq   #0,d0
    lea     graf_name(pc),a1; Graphics-Library

    CALLEXEC     OpenLibrary ; öffnen
    tst.l   d0
    beq     quit
    move.l  d0,_GfxBase
    lea     NewScreen(pc),a0; Screen öffnen
    CALLINT OpenScreen
    tst.l   d0
    beq     quit
    move.l  d0,Screen
    lea     NewWindow(pc),a0; Window öffnen
    CALLINT OpenWindow ;  (dadurch wird der Screen
    tst.l   d0         ;   gelöscht!)
    beq     closescr
    move.l  d0,Window
    lea.l   Font(pc),a0;Zeiger auf TOPAZ-80 holen
    CALLGRAF          OpenFont
    move.l  d0,FONT

    lea.l   TEXT,a5      ; Scroll-Text
    moveq.l #0,d5        ; Scroll-Zähler (0-7)

R1  and.l   #$7,d5
    bne     R4           ; neuer Buchstabe fällig? ?
    move.l  FONT,a2
    move.l  34(a2),a2    ; Zeiger auf Font-Daten
    add.l   #7*$c0,a2    ; unterste Zeile der
                           Zeichendaten
    moveq.l #0,d0
R6  move.b  (a5)+,d0     ; Nächster Buchstabe
    bne     R5           ; Wenn letzter,  von vorne
    lea.l   TEXT,a5      ; anfangen
    bra     R6
R5  cmp.b   #$7f,d0
    bls     R9
    sub.b   #$20,d0
R9  sub.b   #$20,d0      ; Zeiger auf Font-Daten des
    add.l   d0,a2        ; Buchstabens richten
R4  or.b    #$8,d5       ; Scroll-Zähler um 8
                           erhöhen, da Daten
                         ; in Byteform vorliegt,
                           rolx aber nur
                         ; Word zuläßt !
    move.l  Screen,a1    ; Zeiger auf Bitplane 1
    move.l  $c0(a1),a0
    add.l   #8*40,a0     ; Zeiger auf unterste
                         ; Textzeile richten
    move.l  #7,d1        ; 7+1 Zeilen scrollen
    move.l  a2,a1
R7  move.w  $dff006,d0 ; DMA-Position abwarten
    lsr.w   #8,d0        ; Y-Werte herausfiltern
    cmp.b   #$50,d0
    bls.s   R7
R2  move.l  #19,d0       ; 19+1 Bytes scrollen
    moveq   #0,d2        ; Bit des reinscollenden
    move.b  (a1),d2      ; Zeichens ermitteln
    roxl.w  #1,d2
    roxl.w  d5,d2
    sub.l   #$c0,a1
R8  roxl.w  -(a0)        ; Bytes scrollen
    dbra    d0,R8
    dbra    d1,R2
    move.l  #$e00,d0     ; kurze Pause
R3  dbra    d0,R3
    addq    #1,d5
    btst    #6,$bfe001 ; Maustaste gedrückt?
    bne     R1

closewin     move.l  Window,a0; Window schließen
        CALLINT CloseWindow
closescr     move.l  Screen,a0; Screen schließen
        CALLINT CloseScreen
quit    moveq.l #0,d0
        rts

int_name         INTNAME
graf_name        GRAFNAME
NewScreen
        dc.w     0,0          ; NewScreen-Struktur
        dc.w     320,50
        dc.w     1
        dc.b     0,1
        dc.w     0
        dc.w     CUSTOMSCREEN
        dc.l     0,0,0
NewWindow
        dc.w     0,0          ; NewWindow-Struktur
        dc.w     320,50
        dc.b     0,1
        dc.l     GADGETDOWN!DISKINSERTED!
                 DISKREMOVED
        dc.l     SMART_REFRESH!ACTIVATE!RMBTRAP!
                 BORDERLESS
        dc.l     0
        dc.l     0
        dc.l     0
Screen  ds.l     1
        dc.l     0
        dc.w     100,20,640,200
        dc.w     CUSTOMSCREEN

_IntuitionBase   dc.l    0

_GfxBase         dc.l    0
Window  dc.l     0
Font    dc.l     fontname     ; Font-Struktur zum
                                Öffnen
        dc.w     TOPAZ_EIGHTY
        dc.b     FS_NORMAL,FPF_ROMFONT
fontname
        dc.b     "topaz.font",0

FONT    dc.l     0
* eigentlicher Scroll-Text, mit 0 terminiert! !
TEXT    dc.b     "KICKSTART-Die AMIGA-Zeitschrift
                  für den AMIGA-Freak, "
        dc.b     "der mehr über seine
                  Wundermaschine wissen möchte."
        dc.b     "Für den Kickstartleser ist nun
                  auch ein Scroll-Text kein "
        dc.b     "Geheimnis mehr ......      "
        dc.b     0
* Ende Listing
```

END of KICK

AMIGA BASIC Profibuch



NEU!

Für Theoretiker

Dieses Buch gibt Ihnen einen hervorragenden Überblick über die wichtigsten Möglichkeiten und Methoden der Systemprogrammierung unter AMIGA-BASIC. Neben einer detaillierten Beschreibung und Analyse der wichtigsten Funktionen finden Sie in diesem Buch eine ausführliche Erläuterung der verschiedensten System-Unterroutinen sowie den Aufbau des Gesamtsystems.

Für Praktiker

Im Hinblick auf die praktische Anwendung in eigenen Programmen dürfte die Sammlung von über 60(!) neuen AMIGA-BASIC-Unterroutinen zur Systemprogrammierung ein absoluter Lekkerbissen für jeden BASIC-Programmierer sein. Mit diesen neuen Unterroutinen werden die fantastischen Möglichkeiten des AMIGA nun endlich auch all denjenigen in einfacher Weise zugänglich gemacht, die selbst nicht tiefer ins Sytem eindringen möchten.

Für jeden

Blitzschnelle und variationsreiche Textausgabe, die Benutzung beliebiger Textfonts, die Erzeugung von Pull-Down-Menüs, die Gestaltung eigener Requester sowie die komplette Fenstersteuerung werden ab sofort für jeden BASIC-Programmierer leicht realisierbar.

Aber noch viel mehr Geheimnisse erwarten Sie während der Lektüre der einzelnen Kapitel.

HALLE 7 / E 42
HANNOVER MESSE
CeBIT'89
Welt-Centrum Büro · Information · Telekommunikation
8. – 15. MÄRZ 1989

Bestellcoupon MAXON Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Name: ______________________
Vorname: ______________________
Straße: ______________________
Ort: ______________________
Unterschrift: ______________________

Hiermit bestelle ich:

☐ Exemplar(e) von "AMIGA BASIC Profibuch". Mit Diskette für DM 59,00
☐ Vorauskasse
☐ Nachnahme

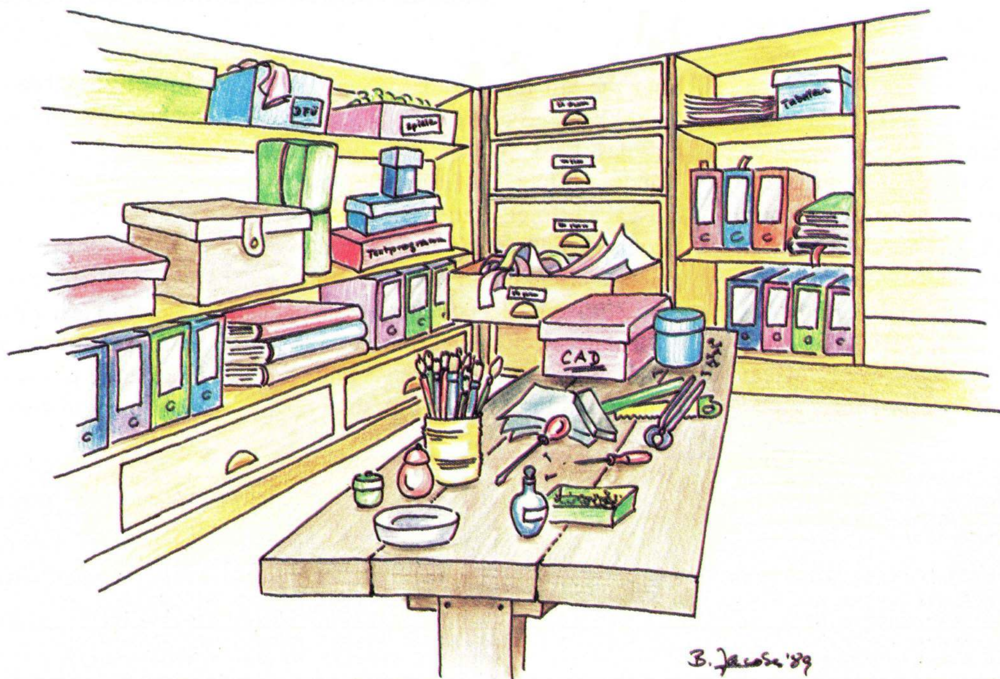
Versandkosten: DM 7,50
Nachnahme zuzgl. DM 3,50 Nachnahmegebühr.

Vertrieb in der Schweiz: DTZ DataTrade AG Langstrasse 94 Postfach 413 CH-8021 Zürich Tel.: 01/242 80 88 Fax.: 01/291 05 07
Vertrieb in Österreich: Dipl.-Ing. Reinhart Temmel Ges.m.b.H. & Co.KG Markt 109 A-5440 Golling Tel.: 06244/7081-17 Fax.: 06244/7188-3
Vertrtieb in Holland: JOTKA COMPUTING Postbus 8183 NL-6710 AD Ede Tel.: 08380/38731 Fax.: 08380/21675

VON THOMAS LIEBMANN

WB ASSIGN

Assign per Workbench



Ist es Ihnen auch schon einmal passiert, daß Sie sich auf der Workbench befanden und nun ein Standard-Device (z.B. FONTS) umlenken oder nur ein logisches Verzeichnis erzeugen wollten? Haben Sie das CLI-Programm nicht auf Ihrer Diskette, oder sind die dafür nötigen Files nicht vorhanden? Dann ist 'WBAssign' genau das richtige für Sie. Mit diesem Programm können Sie, wie vom CLI-Befehl 'Assign' bekannt, die logischen Devices umstellen oder erzeugen.

Voraussetzung: Die CLI-Befehle '*Run*' und '*Assign*' befinden sich im logischen Verzeichnis C: Das Programm wurde mit dem 'Lattice C V3.03' erstellt und kann mit der folgenden Sequenz compiliert werden:

```
LC:lc WBAssign.c
LINK:blink FROM LIB:Lstartup.Fobj+
        WBAssign.o TO
        WBAssign LIB lc.lib+LIB:amiga.lib
```

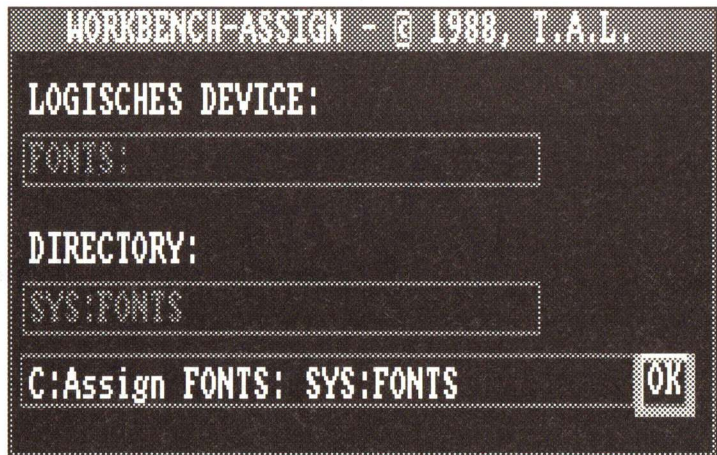
Eventuelle Warnungen beim Compilieren können vernachlässigt werden. Die Anpassung an den Aztec-Compiler dürfte keine Schwierigkeiten bereiten.

Nun brauchen Sie nur noch ein Programm-Icon, (z.B. das CLI-Icon oder ein beliebiges) mit dem Namen 'WBAssign.info' der Zieldatei zu kopieren.

Ein Beispiel:

copy CLI.info to WBAssign.info

Mit diesem Befehlsausdruck erhält das Programm ein Icon, natürlich können Sie auch Icons von anderen Programmen umkopieren. Jedoch darf es kein Ordner sein.



Mit WBAssign können Sie sich das Laden des Command Line Interpreters sparen. Die Zuweisung erfolgt mit der Maus.

Das Programm:

Nun kann das Programm entweder vom CLI (allerdings ziemlich sinnlos, nicht wahr?) oder aber von der Workbench aus gestartet werden. Im erscheinenden Window brauchen Sie nur die entsprechenden Gadgets (die Felder mit dem voreingestellten Text) durch einen Mausklick zu aktivieren und das Feld zu edieren, wie Sie es auch von normalen Workbench-Gadgets (z.B. Rename) gewohnt sind. Dabei ist "Logisches Device" das zu ändernde Device, "Directory" das Zielverzeichnis.

Sind Sie mit Ihren Änderungen fertig, dann klicken Sie mit der Maus einmal auf das Feld "O.K.". Nun wird ein Window öffnet, auf dem die Assign-Ausgabe erfolgt. Hatten Sie die Felder "Logisches Device" und "Directory" geleert, erscheint in diesem Window eine Auflistung der bestehenden Geräte. Nach kurzer Zeit schließt das Programm seine Windows selbsttätig, und Sie können mit Ihrer Arbeit fortfahren.

Auf die Programmierung möchte ich nicht näher eingehen, das Listing ist dokumentiert, so daß Sie wissen, was die einzelnen Programmteile machen.

Der umständliche Weg, 'um 'mal schnell' eine Zuweisung zu machen entfällt mit diesem kleinen Programm. Nach einer gewissen Zeit wollen Sie es nicht mehr missen.

```
/* NAME: WORKBENCH-ASSIGN              *
* (C) MAXON COMPUTER GMBH 1989         *
* AUTOR: THOMAS LIEBMANN               *
* KICKSTART 1989                       */

/*** INCLUDE-FILES & DEFINES    ***/
#include <exec/types.h>
#include <exec/nodes.h>
#include <exec/lists.h>
#include <exec/ports.h>
#include <exec/devices.h>
#include <devices/keymap.h>
#include <graphics/regions.h>
#include <graphics/copper.h>
#include <graphics/gels.h>
#include <graphics/gfxbase.h>
#include <graphics/gfx.h>
#include <graphics/clip.h>
include <graphics/view.h>
#include <graphics/rastport.h>
#include <graphics/layers.h>
#include <hardware/blit.h>
#include <libraries/dos.h>
#include <libraries/dosextens.h>
#include <intuition/intuition.h>
#include <intuition/intuitionbase.h>
#define INT_REV 29

/***         BORDER-DEFINITIONEN          ***/
SHORT Gadget0Vektors[]={ 0,0, 240,0, 240,11,
                          0,11, 0,1 };
struct Border Gadget0Border=
       { -3,-2, 1, 0, JAM1, 5, (SHORT *)Gadget
                               0Vektors,NULL};

SHORT Gadget1Vektors[]={ 0,0, 240,0, 240,11,
                          0,11, 0,1 };
struct Border Gadget1Border=
       { -3,-2, 1, 0, JAM1, 5, (SHORT *)Gadget
                               1Vektors,NULL};

SHORT Gadget2Vektors[]={ 2,0, 22,0, 22,11,
                          2,11, 2,1 };
SHORT MessageVektors[]={ 0,0, 285,0, 285,11,
                          0,11, 0,1 };
struct Border MessageBorder=
       { -290,-2, 1, 0, JAM1, 5,
        (SHORT *)MessageVektors,NULL};
struct Border Gadget2Border=
        { -3,-2, 1, 0, JAM1, 5,
         (SHORT *)Gadget2Vektors,&MessageBorder};

/***  BENÖTIGTE STRUKTUREN & VARIABLEN    ***/
struct GfxBase *GfxBase;
struct IntuitionBase *IntuitionBase;
struct Gadget Gadget[3];
struct Gadget *TestGadget;
struct StringInfo Info[2];
struct Window *GadgetWindow;
struct NewWindow NewWindow;
struct IntuiText Print[3],BoolText;
struct RastPort *rp;
struct IntuiMessage *message;
extern struct FileHandle *Open();
char    From[30]="FONTS:",EndString1[30]=
        "",To[30]="SYS:FONTS",
     EndString2[30]="",Befehl[61]="",Space[2]=" ",
     AssignText[10]="C:Assign ";
long i;
int GadID;
BOOL ende,erfolg;
ULONG MessageClass;
USHORT MessageCode;

/***   BEGINN DES HAUPTPROGRAMMS    ***/
main()
 {

/***    VARIABLEN    ***/
  struct FileHandle *Ausgabe;
  void StringAppend();

/***    EINLESEN DER LIBRARIES    ***/
  IntuitionBase = (struct IntuitionBase *)
  OpenLibrary ("intuition.library",INT_REV);
  if (IntuitionBase == NULL) exit(FALSE);

  GfxBase = (struct GfxBase *)
  OpenLibrary("graphics.library",INT_REV);
  if (GfxBase == NULL) exit(FALSE);

/***  GADGET-DEFINITIONEN    ******/
  Info[0].Buffer=From;
  Info[0].UndoBuffer=EndString1;
  Info[0].MaxChars=30;
  Info[0].BufferPos=0;
  Info[0].DispPos=0;
  Info[0].UndoPos=0;

```

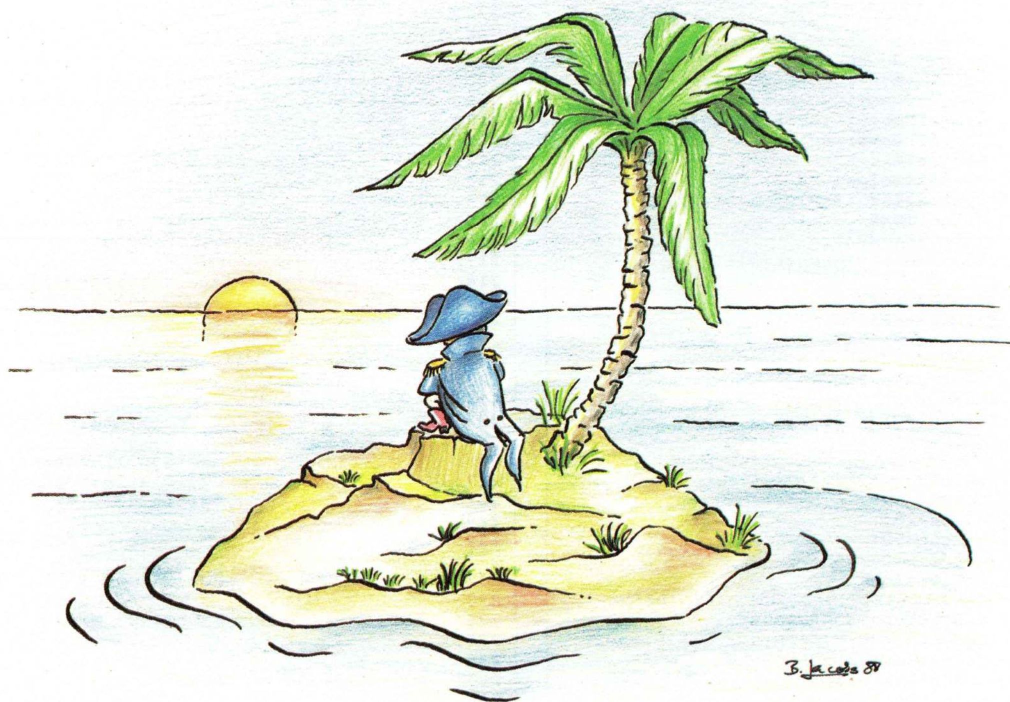
```
    Gadget[0].LeftEdge=10;
    Gadget[0].TopEdge=30;
    Gadget[0].Width=280;
    Gadget[0].Height=10;
    Gadget[0].Flags=GADGHCOMP;
    Gadget[0].Activation=TOGGLESELECT;
    Gadget[0].GadgetType=STRGADGET;
    Gadget[0].GadgetRender=(APTR)&Gadget0Border;
    Gadget[0].SelectRender=NULL;
    Gadget[0].GadgetText=NULL;
    Gadget[0].MutualExclude=NULL;
    Gadget[0].SpecialInfo=(APTR) &Info[0];
    Gadget[0].GadgetID=1;
    Gadget[0].UserData=NULL;
    Gadget[0].NextGadget=&Gadget[1];

    Info[1].Buffer=To;
    Info[1].UndoBuffer=EndString2;
    Info[1].MaxChars=30;
    Info[1].BufferPos=0;
    Info[1].DispPos=0;
    Info[1].UndoPos=0;

    Gadget[1].LeftEdge=10;
    Gadget[1].TopEdge=63;
    Gadget[1].Width=280;
    Gadget[1].Height=10;
    Gadget[1].Flags=GADGHCOMP;
    Gadget[1].Activation=TOGGLESELECT;
    Gadget[1].GadgetType=STRGADGET;
    Gadget[1].GadgetRender=(APTR)&Gadget1Border;
    Gadget[1].SelectRender=NULL;
    Gadget[1].GadgetText=NULL;
    Gadget[1].MutualExclude=NULL;
    Gadget[1].SpecialInfo=(APTR) &Info[1];
    Gadget[1].GadgetID=2;
    Gadget[1].UserData=NULL;
    Gadget[1].NextGadget=&Gadget[2];

    BoolText.ITextFont=NULL;
    BoolText.FrontPen=2;
    BoolText.BackPen=0;
    BoolText.DrawMode=JAM1;
    BoolText.LeftEdge=2;
    BoolText.TopEdge=0;
    BoolText.IText=(UBYTE *) "OK";
    BoolText.NextText=NULL;

    Gadget[2].LeftEdge=296;
    Gadget[2].TopEdge=79;
    Gadget[2].Width=20;
    Gadget[2].Height=10;
    Gadget[2].Flags=GADGHBOX;
    Gadget[2].Activation=
              TOGGLESELECT|GADGIMMEDIATE|RELVERIFY;
    Gadget[2].GadgetType=BOOLGADGET;
    Gadget[2].GadgetRender=(APTR)&Gadget2Border;
    Gadget[2].SelectRender=NULL;
    Gadget[2].GadgetText=&BoolText;
    Gadget[2].MutualExclude=NULL;
    Gadget[2].SpecialInfo=NULL;
    Gadget[2].GadgetID=3;
    Gadget[2].UserData=NULL;
    Gadget[2].NextGadget=NULL;

 /***   WINDOW-DEFINITION   ***/
    NewWindow.LeftEdge=0;
    NewWindow.TopEdge=0;
    NewWindow.Width=330;
    NewWindow.Height=100;
    NewWindow.DetailPen=2;
    NewWindow.BlockPen=1;
    NewWindow.Title=(UBYTE *)
              "WORKBENCH-ASSIGN - (C) MAXON 1989";
    NewWindow.Flags=ACTIVATE|
              SMART_REFRESH|REPORTMOUSE;
    NewWindow.IDCMPFlags=
              REFRESHWINDOW|MOUSEBUTTONS
    |MOUSEMOVE|GADGETDOWN|GADGETUP;
    NewWindow.Type=WBENCHSCREEN;
    NewWindow.FirstGadget=&Gadget[0];
    NewWindow.CheckMark=NULL;
    NewWindow.Screen=NULL;
    NewWindow.BitMap=NULL;
    NewWindow.MinWidth=0;
    NewWindow.MinHeight=0;
    NewWindow.MaxWidth=0;
    NewWindow.MaxHeight=0;
    GadgetWindow = (struct Window *)
                        OpenWindow(&NewWindow);

 /***  TEXT-DEFINITIONEN   ***/
    Print[0].ITextFont=NULL;
    Print[0].FrontPen=2;
    Print[0].BackPen=0;
    Print[0].DrawMode=JAM1;
    Print[0].LeftEdge=10;
    Print[0].TopEdge=17;
    Print[0].IText=(UBYTE *) "LOGISCHES DEVICE:";
    Print[0].NextText=&Print[1];

    Print[1].ITextFont=NULL;
    Print[1].FrontPen=2;
    Print[1].BackPen=0;
    Print[1].DrawMode=JAM1;
    Print[1].LeftEdge=10;
    Print[1].TopEdge=50;
    Print[1].IText=(UBYTE *) "DIRECTORY:";
    Print[1].NextText=NULL;

    Print[2].ITextFont=NULL;
    Print[2].FrontPen=2;
    Print[2].BackPen=0;
    Print[2].DrawMode=JAM1;
    Print[2].LeftEdge=10;
    Print[2].TopEdge=80;
    Print[2].NextText=NULL;

 /***   WINDOW-AUFBAU   ***/
   rp=GadgetWindow->RPort;
   PrintIText(rp,&Print,0,0);
   RefreshGadgets(&Gadget[0],GadgetWindow,NULL);
   ende = FALSE;

 /***   ABFRAGE-SCHLEIFE   ******/
   for (;;)
    {
     if (message=(struct IntuiMessage *)
         GetMsg(GadgetWindow->UserPort))
      {
       MessageClass = message->Class;
       MessageCode  = message->Code;
       ReplyMsg(message);

       switch (MessageClass)
        {
         case GADGETDOWN :
         case GADGETUP   : TestGadget = (struct
         Gadget *)
                  message->IAddress;
                  GadID = TestGadget->GadgetID;
                  switch (GadID)
                            {
                             case 3: ende = TRUE;
                                     break;
                            }
                           break;
         case MOUSEBUTTONS: break;
        }
      }
     if (ende == TRUE)
      {
       StringAppend(Befehl,AssignText);
       StringAppend(Befehl,From);
       StringAppend(Befehl,Space);
       StringAppend(Befehl,To);
       Print[2].IText = (UBYTE *)&Befehl;
       PrintIText(rp,&Print[2],0,0);
       Ausgabe=Open("CON:0/100/640/155/Assign-
                  OutPut",
                  MODE_OLDFILE);
       if ((erfolg=Execute(Befehl,NULL,Ausgabe))
            == FALSE)
            exit(FALSE);
       for (i=0;i<=200000;i++);
       Close (Ausgabe);
       CloseWindow(GadgetWindow);
       CloseLibrary(IntuitionBase);
       CloseLibrary(GfxBase);
       exit (TRUE);
      }
    }
  }

 /***   FUNKTIONEN   ***/
 void StringAppend(s,t)
  register char *s,*t;
  {
   while (*s)
     s++;
   while(*s++ = *t++);
  }
 /* ENDE LISTING */
```

END of KICK

VON FELIX THEISINGER

Make Batch

den Execute-Befehl verbannt



Batchdateien, Befehls- oder auch Stapeldateien genannt, müssen unter AMIGA DOS immer mit dem Befehl 'Execute' gestartet werden. Unter MSDOS ist kein zusätzlicher Befehl vonnöten, das Betriebssystem erkennt diese speziellen Dateien automatisch. Mit dem an dieser Stelle abgedruckten Programm Make-Batch wird ein Kompromiß geschlossen, doch lesen Sie selbst.
"MakeBatch" erzeugt aus jeder beliebigen Batchdatei ein lauffähiges Programm.

Der Aufruf des Programms ist denkbar einfach:
mb <batchdatei>

Die lauffähige Batchdatei mit dem Namen <batchdatei>.bat befindet sich dann in der RAM-Disk.

Execute verbannt!

Der Vorteil des erzeugten Batchdateiprogramms gegenüber der normalen Batchdatei liegt darin, daß man sich den lästigen Befehl *"execute"* sparen kann. Der Anwender braucht dann nicht mehr zwischen Batchdateien, die er mit *"execute"* starten müßte, und normalen Programmen zu unterscheiden. Besonders interessant ist dieses Utilitiy für Fest-

plattenbesitzer, die ihre Programme in einem sehr stark verzweigten, hierachischen Subdirectorysystem gespeichert haben. Sie können dann zu jedem Programm eine Batchdatei, in der alle nötigen Anpassungen (wie *cd*, *assign*, *path*) erledigt werden, und von der das Programms gestartet wird, erstellen. Diese Batchdatei kann mit Hilfe dieses Programm lauffähig gemacht werden. Man braucht dann nur noch den Namen des erzeugten Batchdateiprogramms einzutippen, um ein Programm aus einem tiefergelegenen Subdirectory zu starten.

Zum Programm

Das Programm wurde vollständig in C geschrieben und läßt sich mit dem Aztec-C compilieren (Anpassung an den Lattice-Compiler sollte leicht möglich sein). Da das Programm MakeBatch selbst wieder ein C-Programm erstellt (!) und dieses mit dem Aztec-C compiliert, läuft MakeBatch nur in der Aztec-C-Compiler-Umgebung. Dies bedeutet, daß das Programm MakeBatch nicht von einer normalen Workbench oder anderweitigen Programm-Disketten gestartet werden kann. Weitere Voraussetzungen sind das Vorhandensein der Befehle delete, execute, run sowie die Verfügbarkeit der RAM-Disk.

Die erstellten Batchdateiprogramme sind dann natürlich von diesen Bedingungen unabhängig.

Funktionsweise

MakeBatch liest die Batchdatei, die als Parameter übergeben wurde, vollständig ein und speichert diese in dem neu erstellten File *"ram:bat.c"* in C-Syntax als char-Array ab. Danach wird das C-Programm, das in MakeBatch.c im Array char **prg[]* gespeichert ist, in *"ram:bat.c"* geschrieben. MakeBatch ruft dann den Aztec C-Compiler auf und compiliert *"ram:bat.c"*. Das Kompilat *"ram:bat"* wird in <name>.bat umbenannt, und die Files *"ram:bat#?#"* werden gelöscht.

Die von *". key "* geforderten Parameter werden vom Programm aus durch die Übergabeparameter *argv[1] geliefert.

Für Tippfaule ist sowohl der Sourcecode als auch das fertige Programm von unserm Disketten-Servive erhältlich.

```
/*    Makebatch                       *
 *    Autor: Felix Theisinger         *
 *    (c) MAXON Computer GmbH         *
 *    1989 KICKSTART                  */

#include <stdio. h>

char *prg[] =           /* C-PRG Quellcode */
{
   "void main(argc, argv)\n",
   "int  argc; \n",
   "char *argv[]; \n{\n",
   "   char p[80], *argu[3]; \n",
   "   int  handle, i, j; \n\n\n",
   "   handle  = creat(\"ram:batch\", 0L);\n",
   "   write(handle, batch, sizeof(batch));\n",
   "   close(handle); \n",
   "   argu[0] = argu[2] = 0L; \n",
   "   argu[1] = p; \n",
   "   strcpy(p, \"ram:batch\"); \n",
   "   for(i=1;  i<argc;  i++)\n   {\n",
   "      p[(j = strlen(p))+1] = 0; \n",
   "      p[j] = ' '; \n",
   "      strcat(p, argv[i]); \n   }\n",
   "   fexecv(\"execute\", argu); \n",
   "   DeleteFile(\"ram:batch\"); \n}\n",
};

void main(argc, argv)
int  argc;
char *argv[];
{
   FILE *eing, *ausg, *fopen();
   int  c, z=0;
   char buffer[40];
   puts("MakeBatch von Felix Theisinger (24.7.
   88) \n");
   if(argc!= 2)
   {
      puts("\nSyntax: \nmakebatch batchdatei\n");
      return;
   }
   eing = fopen(argv[1],  "r");
   ausg = fopen("ram:bat. c",  "w");
   if(eing == NULL || ausg == NULL)
   {
      puts("\nfile error!!! \n");
      return;
   }

   fputs("char batch[] = \n{\n",  ausg);
 /* Batch-Datei als char-Array anlegen */
   while(1)
   {
      c = fgetc(eing);
      if(c == -1)
         break;
      fprintf(ausg,  " %3d, ", c);    putchar(c);
      if(z++ > 8) {
        fputc('\n', ausg);
        z=0;
      }
   }
   fputs("\n}; \n\n", ausg);

   for(z=0;  z < 17;  z++)      /* C-Programm
                                  abspeichern */
      fputs(prg[z],  ausg);
   fclose(eing);
   fclose(ausg);
   Execute("cc >nil: ram:bat. c", 0L, 0L);/*
   C-PRG compilieren */
   Execute("ln >nil: ram:bat. o -lc", 0L, 0L);
   strcpy(buffer, argv[1]);  strcat(buffer, ".
   bat");
   Rename("ram:bat", buffer);
   Execute("delete ram:bat#?# q", 0L, 0L);
}
```

Listing zum Programm MAKEBATCH

END of KICK

VON FRANK SCHÄFER

Geheime Infos

Zwei Tips zum Umgang mit der Workbench

Bei dem Handling mit der Workbench und dem CLI gibt es (unter anderem) zwei grundlegende Probleme: Wie erhält man auf eine einfache Art und Weise den Screenpointer auf den Workbench- bzw. CLI-Screen, und wie kann festgestellt werden, ob ein Programm von der Workbench oder vom CLI aus gestartet wurde?

Zur Lösung des zweiten Problemes habe ich ein kleines Modul mit dem Namen "*FromWB*" programmiert, das dem Modula 2-Programmierer eine Prozedur mit dem Namen "*StartFromWB*" zur Verfügung stellt. Diese Prozedur liefert TRUE zurück, falls das Programm von der Workbench aus gestartet wurde.

GetLock

Die Routine *GetLock*, die aus dem Modul *Arguments* importiert wird hat an sich eine andere Aufgabe, als mit ihr festzustellen, ob der Start von der Workbench aus erfolgte. Sie liefert, falls das Programm von Workbench aus gestartet wurde, den *FileLockPtr* auf die übergebenen Argumente (Files & Programme) zurück. Falls man Null (0) als Parameter an *GetLock* übergibt, erhält man den *FileLockPtr* auf das eigentlich Programm zurück.

Der Screenpointer

Den Screenpointer auf den Workbench- bzw. CLI-Screen bekommt man durch den Aufruf der Routine "*OpenWorkBench ()*". Diese Routine gibt den Pointer auf den Workbench-Screen zurück. Falls die Workbench mit "*CloseWorkBench ()*" geschlossen war, wird sie wieder geöffnet. Wenn sie schon geöffnet war, passiert nichts weiter, als daß man den Pointer auf den Workbench-Screen zurückbekommt. Die Kenntnis des Pointers erlaubt einem viele Dinge; zum Beispiel kann man die Farben der Workbench verändern, neue Zeichen-

sätze laden oder einige lustige Dinge mit den Windows, die sich auf dem Screen befinden, anstellen.

Hier sollte noch einmal gesagt werden, daß der Aufruf der Routine "*OpenWorkBench ()*" nichts mit dem CLI-Befehl *LoadWB* zu tun hat. Auch wenn die Workbench mit *LoadWB* (noch) nicht geladen wurde, erhält man einen Screen-Pointer als Resultat der OpenWorkBench-Routine. Das CLI-Window und die Workbench liegen ja bekanntlich auf dem gleichen Screen...

In dem Beispiel wird mittels der Funktion *OpenWorkBench ()* der Screen-Pointer bestimmt. Die Intuition-Procedure *MoveScreen* erlaubt es, einen Screen nach oben und unten zu bewegen. In den beiden Schleifen wird der Workbench-Screen erst langsam nach unten und dann doppelt so schnell wieder nach oben bewegt.

```
IMPLEMENTATION MODULE FromWB;

(* FS *)

FROM Arguments  IMPORT  GetLock;

PROCEDURE StartFromWB (): BOOLEAN;

BEGIN
 IF GetLock (0) = NIL
THEN
   RETURN FALSE
 END;
 RETURN TRUE
END StartFromWB;

END FromWB.
```

```
DEFINITION MODULE FromWB;

(* Feststellen, ob ein Programm von Workbench
   gestartet  *)
(* wurde. Falls das der Fall ist, liefert die
   Procedure  *)
(* StartFromWB ein TRUE zurück. *)
(* Autor: Frank Schäfer         *)
(* (C) MAXON Computer GmbH 1989 *)
(* KICKSTART 1989               *)

PROCEDURE StartFromWB (): BOOLEAN;

END
FromWB.
```

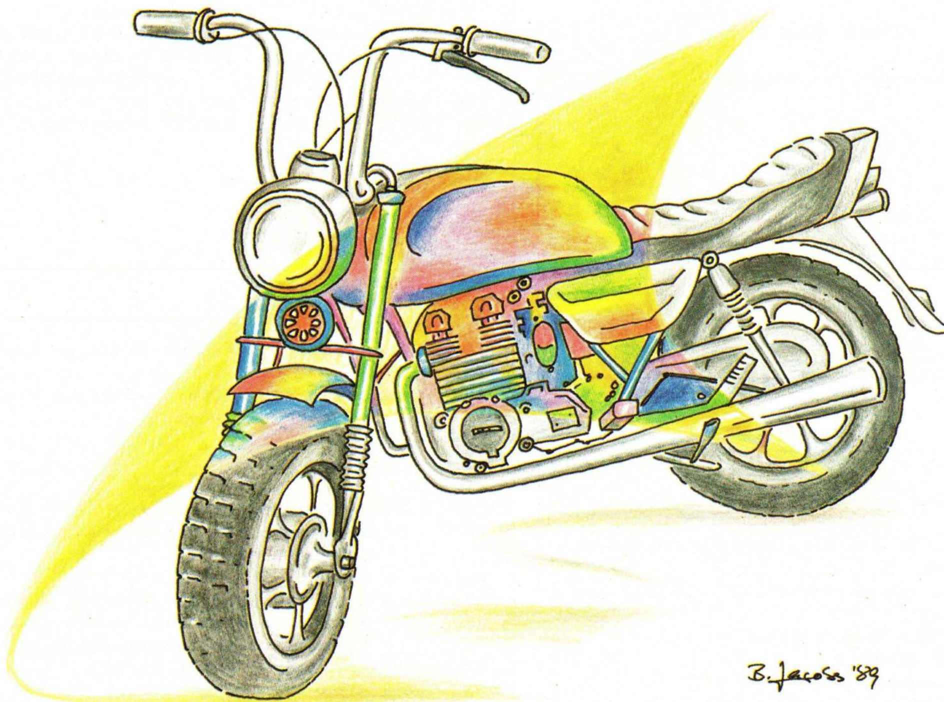
```
MODULE TestFromWB;

(* FS *)
FROM Arts         IMPORT Assert;
FROM FromWB       IMPORT StartFromWB;
FROM Intuition    IMPORT ScreenPtr,MoveScreen,
                         OpenWorkBench;
FROM SYSTEM       IMPORT ADR;
FROM Terminal     IMPORT WriteString,WriteLn;

VAR SPtr:
ScreenPtr;
    Ok   : BOOLEAN;
    i    : INTEGER;

BEGIN
 IF StartFromWB ()
THEN
   WriteString ("Start erfolgte von WorkBench ")
 ELSE
   WriteString ("Start erfolgte nicht von
                 WorkBench ")
 END;
 WriteLn;

 SPtr:= OpenWorkBench ();
 (* So bekommt man den ScreenPtr auf die
    Workbench *)

Assert (SPtr # NIL,ADR("Wo ist die Workbench?"));
  (* Workbench ließ sich nicht öffnen.
     Vermutlich Speichermangel. *)
 WriteString (" Don't Panic!");
 WriteLn;
 FOR i:= 0 TO 255 DO      (* Nach unten ... *)
   MoveScreen (SPtr,0,1)
 END;
 FOR i:= 0 TO 255 BY 2 DO (* Nach oben *)
   MoveScreen (SPtr,0,-2)
 END;
END TestFromWB.
```

END of KICK

VON JULIAN RAUTENBERG

Volle Lotte

2048 Farben für den Copper



Die meisten Leute haben schon was vom Copper gehört. Viele halten die Copperprogrammierung für viel zu kompliziert (so ging es mir zumindest früher). Es gibt jedoch eine recht einfache Möglichkeit, den Copper zu nutzen. Wie? Lesen Sie weiter.

Der Copper

Der Copper besitzt nämlich einen Zeiger auf eine Usercopperliste. Diese Liste kann aufgestellt und dann dem Copper übergeben werden. Nun aber mal von vorne. Der Copper ist ein Coprozessor, der weiß, wo sich der Elektronenstrahl auf unserem Bildschirm befindet, und hat die Aufgabe, zu bestimmten Zeiten die Werte von Registern zu verändern. Er ermöglicht es, mehrere Screens mit verschiedenen Bitplanes darzustellen.

Die drei Copperbefehle

Es gibt drei Libraryfunktionen in der *Graphics.library*, mit denen man eine Usercopperliste programmieren kann:

Cbump	-366
Cmove	-372
Cwait	-378

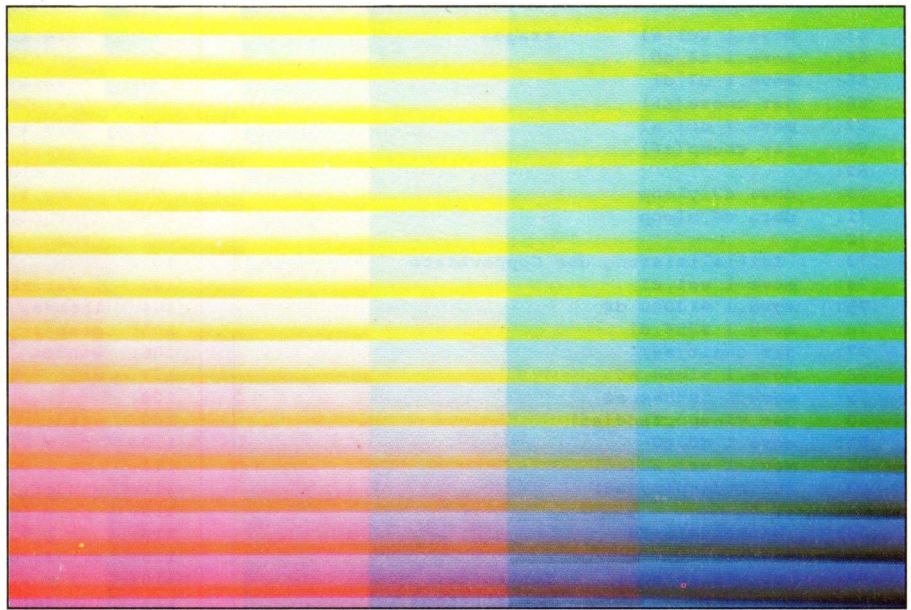
Cwait wartet auf eine Position auf dem Bildschirm und braucht drei Parameter. Sie braucht, wie jede der Funktionen, die Adresse der Copperliste in a0, die Y-Koordinate in d0 und die X-Koordinate in d1.

Cmove schreibt den Wert aus d1 in das Register aus d0, wobei nur der Offset benötigt wird (Basis ist dff000). Auch diese Funktion benötigt in a0 die Addresse der Copperliste.

Cbump benötigt nur die Addresse der Copperliste in a0.

2048 Farben mit einer Bitplane

Das Programm benutzt die Fähigkeiten des Coppers, um die Hintergrundfarbe zu wechseln. Es erscheinen so 2048 Farben, obwohl der Screen nur 1 Bitplane hat und die Vordergrundfarbe nicht benutzt wird. Am Anfang wird der Speicher für die Copperliste reserviert. Dieser Bereich wird mit dem Schließen des Windows wieder freigegeben.



Das Programm zeigt Ihnen, wie leicht es möglich ist, mit nur einer Bitplane 2048 Farben darzustellen.

Anschließend werden die Libraries , der Screen und das Window geöffnet. Um die Copperliste aufzubauen, benutze ich eine Schleife. Das heißt, genauer gesagt, zwei: eine für jede Zeile und eine für die 256 Zeilen. Am Anfang der Schleife errechnet das Programm die richtigen Werte. Es sieht zwar ziemlich umständlich aus, ist jedoch schnell. In den letzten Zeilen der Schleife werden die Funktionen aufgerufen. Um die Copperliste zu aktivieren, wird noch ein Befehl angefügt, der den Copper auf einen nie zu erreichenden Punkt warten läßt, und die Rethink-Display-Funktion aufgerufen, die die Copperliste einbindet. Vorher muß jedoch die Copperliste noch eingetragen werden. Die dafür nötige Adresse des Viewports bekommt man durch die Viewportadress-Funktion. Damit man das Werk betrachten kann, habe ich eine Warteschleife eingebaut, die auf das Drücken des Mausknopfes wartet. Dann werden die Libraries, der Screen und das Window wieder geschlossen.

Das kleine Programm wurde mit dem Seka-Assembler erstellt, eine Anpassung an andere Assembler dürfte aber mit den jetzt erhaltenen Informationen keine Schwierigkeit sein. Das Listing ist des weiteren noch gut dokumentiert.

```
; Copper Programmierung
; Autor: Julian Rautenberg
; (C) MAXON Computer 1989
; KICKSTART 1989

allocmem = -198
openlib = -408
closelib = -414
openscr = -198
openwin = -204
closescr = -66
closewin = -72
rtdisplay = -390
viewportadd = -300
cbump = -366
cmove = -372
cwait = -378

 move.l 4,a6          ; Speicher-
 move.l #65538,d1     ; reservierung
 moveq #12,d0         ; für User-
 jsr allocmem(a6)     ; Copper-Liste
 move.l d0,ucl
 lea intnam,a1
 jsr openlib(a6)
 move.l d0,intbas
 lea granam,a1
 jsr openlib(a6)      ; öffnen der
 move.l d0,grabas     ; Libraries,
 move.l intbas,a6     ; des Screens
 lea scrdef,a0        ; und des
 jsr openscr(a6)      ; Windows
 move.l d0,scrhd
 lea windef,a0
 jsr openwin(a6)
 move.l d0,winhd

; Aufbauen der CopperListe
 move.l grabas,a6
 move.l #256,d4       ; y dbra-Wert
 move.l #384,a4       ; Farbregister
xloop:
 moveq #7,d3          ; x dbra-Wert
yloop:
 moveq #7,d2          ; Errechnung
 sub.l d3,d2          ; der Farbe in
 move.l d2,d7         ; d7, des X-
 asl #8,d7            ; Werts in d1,
 asl #1,d7            ; des y-Werts
 move.l d2,d1         ; in d0
 asl #4,d1
 sub.l d2,d1
 asl #3,d2
 add.l d2,d1
 addi.l #45,d1
 move.l #256,d0
 sub.l d4,d0
 add.l d0,d7
 move.l ucl,a1
 jsr cwait(a6)        ; Wait und move-
 move.l ucl,a1        ; Befehl in
 jsr cbump(a6)        ; Liste über-
```

```
  move.l ucl,a1      ; tragen
  move.l a4,d0
  move.l d7,d1
  jsr cmove(a6)
  move.l ucl,a1
  jsr cbump(a6)

  dbra d3,yloop
  dbra d4,xloop

 ; Initialisierung der CopperListe
  move.l ucl,a1
  move.l #10000,d0
  move.l #256,d1
  jsr cwait(a6)
  move.l winhd,a0
  move.l intbas,a6
  jsr viewportadd(a6)
  move.l d0,a0
  move.l ucl,20(a0)
  jsr rtdisplay(a6)

 ; Warteschleife
 wait:
  btst #6,$bfe001    ; Testen des
  bne wait           ; Mousebuttons

 ; Ende
  move.l intbas,a6
  move.l winhd,a0    ; Schließen der
  jsr closewin(a6)   ; Librarys, des
  move.l scrhd,a0    ; Screens und
  jsr closescr(a6)   ; des Windows
  move.l 4,a6
  move.l intbas,a1
  jsr closelib(a6)
  move.l grabas,a1
  jsr closelib(a6)
  rts

 ; Variablen und Strukturen
 intbas: dc.l 0
 grabas: dc.l 0
 winhd:  dc.l 0
 ucl:    dc.l 0

 scrdef:
  dc.w 0,0,640,266,1,0,$8002,15
  dc.l 0,0,0,0
 windef:
  dc.w 588,0,52,10,0
  dc.l 0,$10000 ,0,0,0
 scrhd: dc.l 0
  dc.w 0,0,0,0,52,20,15
 intnam: dc.b "intuition.library",0
 granam: dc.b "graphics.library",0,0
 ;Listing ENDE
```

END of KICK

KICKS FÜR INSIDER

Liebe Leser, KICKS für INSIDER ist eine Rubrik, die für interessante und lehrreiche Programme geschaffen wurde, denn was nutzt das schönste Programm, wenn man es nur abtippen, aber nicht verstehen kann.

Um jeden Monat neue KICKS zu veröffentlichen, sind wir natürlich auch von unseren Lesern abhängig, denn irgendwann einmal sind die Einfälle jedes Programmierers erschöpft. Also, nehmen Sie an dieser Rubrik aktiv teil, ergänzen Sie sie mit Ihren eigenen Programmen. Werden Sie Autor in der KICKSTART.

Dabei sind alle auf dem AMIGA existierenden Programmiersprachen gewünscht. Egal, ob Basic, C, Assembler, Modula 2, Pascal oder andere.

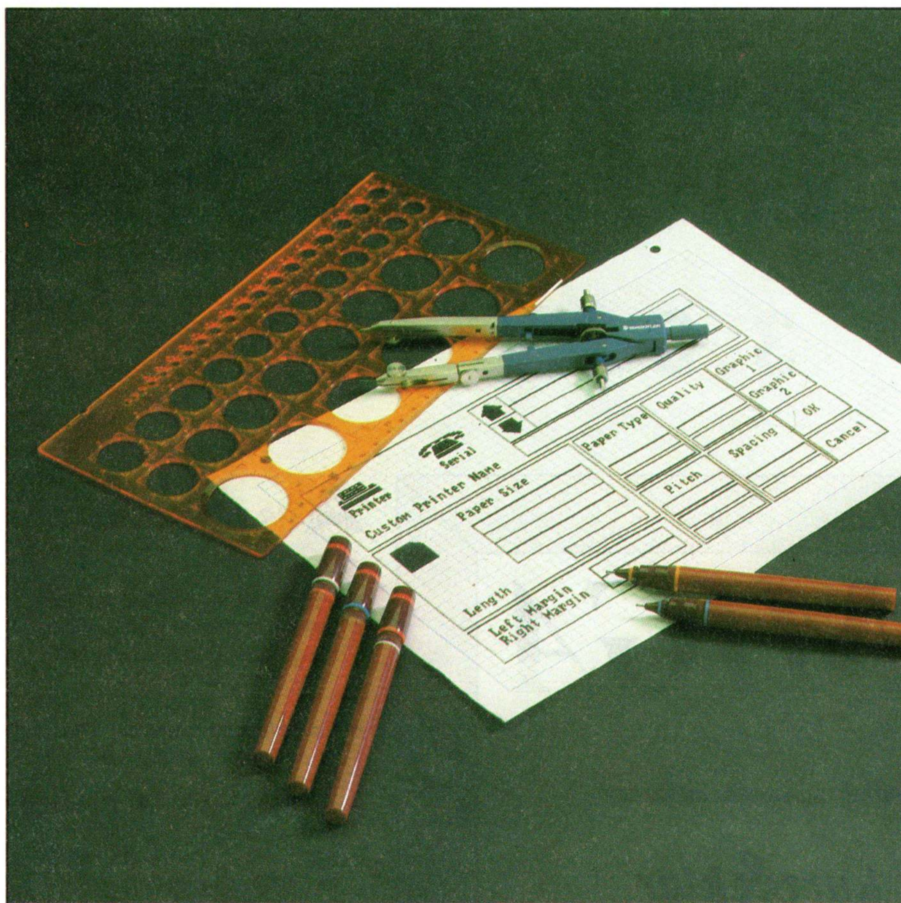
WIE DAS GEHT?

Ganz einfach. Schicken Sie uns das Programm, samt dem zugehörigen Artikel, eventuellen Bildern und Hardcopies (nach Möglichkeit "gegrabbt") auf Diskette zu. Wir werden uns dann bei Ihnen melden. Ein interessantes Honorar wartet bei Veröffentlichung natürlich auch auf Sie.

Wenn Sie Fragen haben, rufen Sie doch einfach bei uns an.

Tel. 06196/481811
Mo-Fr. 11 bis 17 Uhr

Ihre KICKSTART-Redaktion



Im Bereich der Programmiertools gibt es für den AMIGA die verschiedensten Programme wie Debugger, Editoren usw.. Aber Tools zur Erleichterung der Oberflächengestaltung sind sehr selten und meist nicht da einzusetzen, wo sie benötigt werden. Das neu entwickelte R.C.T. schlägt alle um Längen.

INTUITION leicht gemacht

Das R.C.T. in der Praxis

Welcher Programmierer kennt sie nicht, die komplexen Strukturen, die es zu füllen gilt, um auch nur ein Fenster oder Requester auf dem Bildschirm erscheinen zu lassen? Basic-Programmierer wünschen sich wenigstens diese Möglichkeit aber weder AMIGA-BASIC noch das neue GFA-Basic stellen eine ausreichend komfortable Schnittstelle zur Benutzeroberfläche bereit.

Das R.C.T. schließt diese Lücke. Requester und Menüleisten können komfortabel mit der Maus konstruiert werden. Durch die Funktionen der mitgelieferten AMIGA-Library ist es möglich, die erzeugten Requester sogar in BASIC-Programme einzubinden.

Lieferumfang

Bei allen Tools, die einem die Programmierung erleichtern sollen, ist die Dokumentation ein wichtiger Bestandteil.
Der Titel R.C.T. steht für Requester Construction Tool, die Anleitung und das Programm jedoch ist in deutscher Sprache verfaßt, ca. 90 Seiten stark und läßt den User mit dem komplexen Thema nicht alleine.

Es werden nicht nur alle Menüs, sondern auch alle Programmfunktionen leicht verständlich, wenn auch nicht gerade ausführlich, erläutert.

Die auf der R.C.T.-Diskette vorhandenen Programme sind in der Einleitung mit einer kurzen Funktionsbeschreibung aufgeführt.

Die wichtigsten Programme:

RCT das Hauptprogramm zur Gestaltung von Requestern und Menüleisten

RCT_TO_ASCII der Source-Code-Generator. Er generiert GFA-BASIC-, C- und Assembler-Source aus den vom Hauptprogramm erzeugten Dateien

RCT.LIBRARY Eine AMIGA-Library für den Libs-Ordner der Bootdiskette.
Sie kann mit der Betriebssystemfunktion OpenLibrary geöffnet und mit ihren Funktionen zur Verwaltung der Requester und Menüleisten verwendet werden.

Des weiteren sind einige Demos auf der Diskette, die einem den Umgang mit den Library-Funktionen und den erzeugten Source-Codes verdeutlichen sollen.

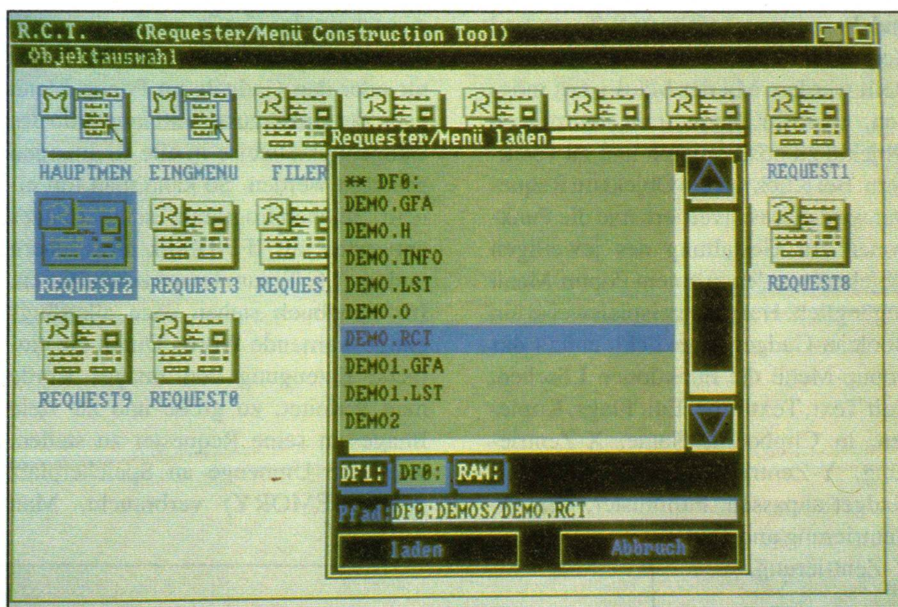


Abbildung 1: Die fertigen Requester und Menüleisten werden als Images angelegt.

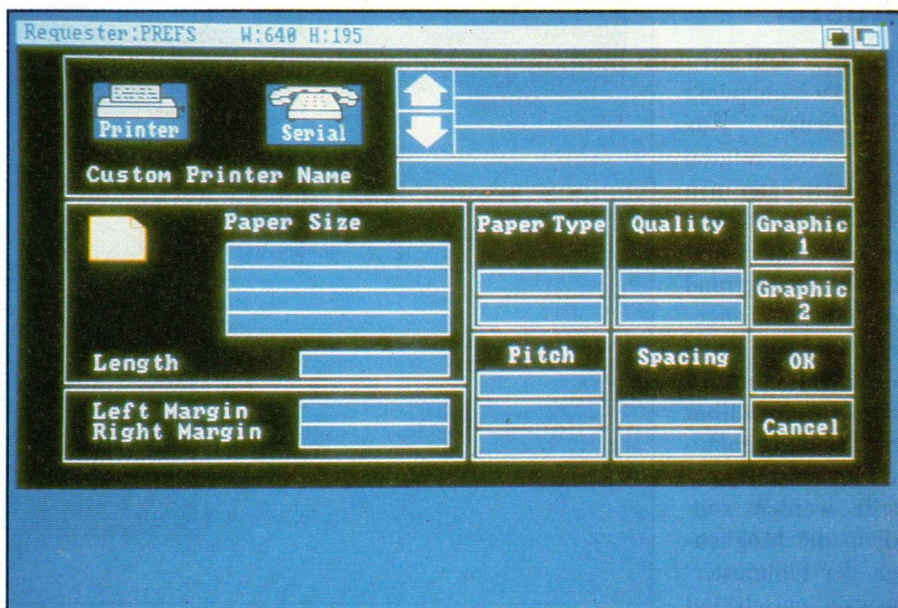


Abbildung 2: Eine knappe viertel Stunde dauert die Nachbildung des Preferences-Requesters. Der daraus erzeugte Source-Code umfaßt ca. 700 Zeilen. Das Erzeugen eines solchen Requesters ohne R.C.T. wäre eine wochenendfüllende Tätigkeit.

Der erste Kontakt

Für Tools durchaus nicht selbstverständlich ist die Möglichkeit des Starts aus der Workbench. Beim R.C.T. ist dies möglich. Es wird sogar für jede erzeugte R.C.T.-Datei ein Icon mitgespeichert. Mit Klick auf ein solches Icon wird das R.C.T. und die angeklickte R.C.T.-Datei geladen. Das Programm öffnet einen eigenen Screen und ist somit nach dem Start nicht zu übersehen.
In diesem Programmteil können Requester und Menüleisten gespeichert, geladen, gelöscht und neu hinzugefügt werden. Die einzelnen Funktionen sind über die Menüleiste des Objektauswahlfensters erreichbar, in der die Menüs Info, Datei und Objektbearbeitung enthalten sind. Beim Laden von R.C.T.-Dateien erscheint ein File-Requester, aus dem die gewünschte Datei ausgewählt werden kann. Lädt man eine R.C.T.-Datei, erscheint nach kurzer Zeit deren Inhalt in Form von "Requester- und Menü-Images" auf dem Fenster.
Unter den Images stehen die Namen der Objekte die beim Neukonstruieren eingegeben werden müssen. Klickt man mit der Maus auf eines der Images, wird es invertiert und bei nochmaligem Klick, je nach gewähltem Objekt, wird entweder in den Menü- oder Requester-Editor verzweigt.

Ungewöhnliche Menüführung

Im Menü- wie auch im Requester-Editor werden sogenannte Popup-Menüs eingesetzt. Im Unterschied zu normalen Menüs erscheinen "Popup's" direkt, wenn die rechte Maustaste gedrückt wird, an der aktuellen Mausposition. Zu Anfang ist diese Menüführung sehr lästig, da man gewohnt ist, nach dem Drücken der rechten Maustaste mit der Maus in die Screen-Titelleiste zu fahren, um ein Menü herunterzuklappen. Bei "Popup's" muß die Maus nur sehr kurze Distanzen über den Bildschirm bewegt werden, was die Auswahl der Menüeinträge beschleunigt. Nur schade, daß im Menü-Editor keine Popup-Menüs konstruiert werden können!

Der Requester-Editor

Nach einem Klick auf ein Requester-Image im Objektauswahlfenster wird der Requester mit all seinen Gadgets und Texten auf dem Bildschirm angezeigt. Über das Requester-Popup-Menü können neue Gadgets (Boolean-, String-

und Proportional-Gadgets), Texte und Requester-Border in den Requester gestellt werden. Mit der Maus ist es möglich, alle im Requester angezeigten Objekte zu positionieren und zu verändern. Bei Klick auf ein Objekt im Requester wird dieses invertiert, und die Funktionen zur Gestaltung des jeweiligen Objektes werden in einem Popup-Menü zugänglich. Hat man beispielsweise ein Boolean-Gadget angeklickt, enhält das Popup-Menü die Funktionen Löschen, Edit Text, Textfarbe, Edit Flags, Kopieren, in Clipboard, Name, X-Zentrierung, Y-Zentrierung, Gadget-Border, Gadget anpassen, Füllmuster, Text X-Zentrierung und Text Y-Zentrierung.

Füllmuster

Einige Neuheiten bietet der Menüpunkt "Füllmuster". Aus dem angezeigten Requester kann man eines von 36 Füllmustern auswählen, das dann im Hintergrund des Gadgets angezeigt wird. Die Farbe des Füllmusters ist ebenso einstellbar wie der Zeichenmodus, in dem es dargestellt werden soll. Allein die Möglichkeit der Füllmusterauswahl entschädigt für das fehlende selbstdefinierbare Füllmuster. Die gestalterischen Effekte die dadurch erzielt werden können sind ähnlich denen von Images in Gadgets, aber bei weitem nicht so speicheraufwendig.

Apropos Images: Hierfür haben sich die Programmierer eine sehr schöne Lösung einfallen lassen. Im R.C.T. können Ausschnitte oder auch ganze IFF-Grafiken für ein Image verwendet werden. Das Popup-Menü im Image-Editor enthält Einträge zum Laden und Auswählen von IFF-Grafiken. Lädt man eine IFF-Grafik, wird ein Ausschnitt daraus, entsprechend der Größe des Gadgets, innerhalb dessen angezeigt. Vergrößert man das Gadget, vergrößert sich der Ausschnitt der Grafik automatisch mit. Aber damit nicht genug. Mit der Tastatur oder den Gadgets im Popup-Menü kann der Grafikausschnitt innerhalb des Gadgets pixelweise in alle Richtungen gescrollt werden. So kann man mit einem Zeichenprogramm seine Images entwerfen, als IFF-Grafik abspeichern und im R.C.T. zum Image konvertieren. Im Handbuch stehen dazu allerdings einige warnende Worte. Durch die einfache Erzeugung von Images würde man verleitet, zu große und zu viele Images in seine Requester zu stellen, was eine Unmenge an Speicherplatz (CHIPMEMORY) verbraucht. Man sollte daher mit der Anzahl der Bilder und der Tiefe der Bitplanes sparsam umgehen.

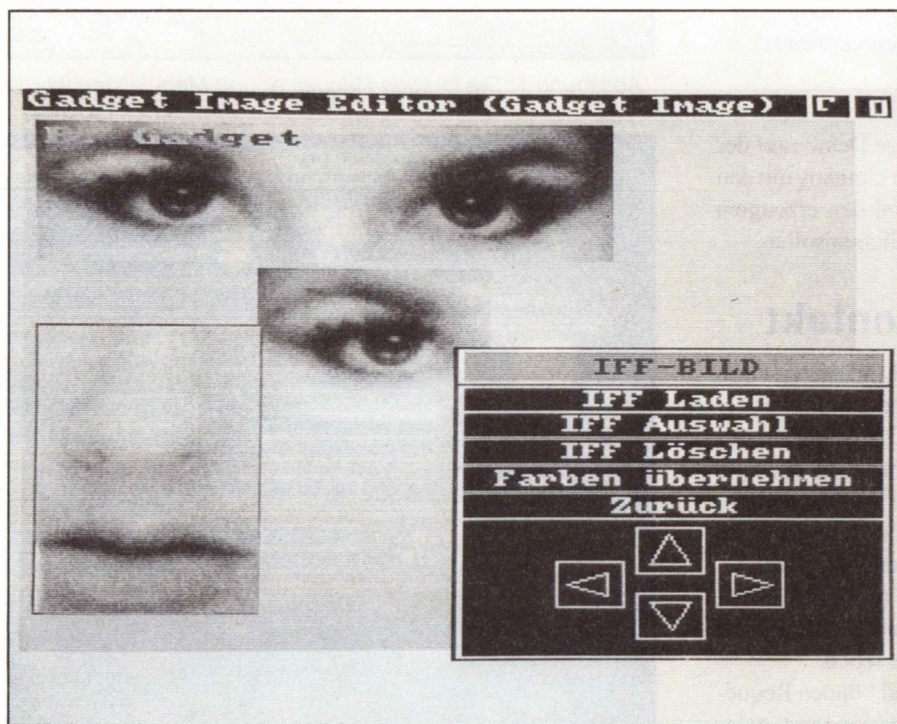


Abbildung 3: Mit dem R.C.T. lassen sich auf einfachste Weise Ausschnitte aus IFF-Bildern in das eigene Programm einbinden.

Gadget- und Requester-Border

Border sind aneinanderhängende Linien. Sie begrenzen Gadgets oder dienen im Requester als Flächenteiler. Anders als bei Requester-Bordern ist die Position eines Gadget-Borders relativ zu der des Gadgets. Wird ein Gadget verschoben, verschiebt sich der Border automatisch mit. Zum Verschieben eines Requester-Borders muß mit der Maus unbedingt ein Eckpunkt getroffen werden, was bei komplexen Bordern etwas Geduld erfordert. Sehr schön sind die Funktionen zum Gestalten von Requester- und Gadget-Bordern. Es können sogar, durch den Symmetriemodus der Borderfunktionen, exakte runde Border erzeugt werden.

Der Menüeditor

Ein weiterer Programmteil ist der Menü-Editor. In dem Fenster, das sich öffnet, wenn ein Menü-Image aus der Objektauswahl angeklickt oder ein neues Menü hinzugefügt wird, steht in der ersten Zeile die Titelleiste des Menüs. Wird mit der Maus einer der Titel angeklickt, klappt das zum Titel gehörende Pull-Down-Menü herunter. Das Beschriebene funktioniert natürlich erst bei einer bereits konstruierten Menüleiste. Beim "neu Hinzufügen" steht lediglich der erste Menütitel in der Titelleiste. Über die Funktionen der Popup-Menüs können neue Titel, Einträge und Untereinträge angehängt werden.

Menüeinträge und Untereinträge

Klickt man auf einen Titel, invertiert er sich, und man kann Menüeinträge an den Titel anhängen. Jeder angehängte Menüeintrag und Untereintrag wird direkt sichtbar, da das Menü vom R.C.T. simuliert wird. Das bedeutet, man sieht immer die komplette Menüleiste so vor sich, wie sie später im eigenen Programm bei Klick auf die rechte Maustaste erscheinen wird.
Wird ein Menüeintrag oder ein Untereintrag (Submenü) eines Menüeintrages angeklickt, kann dieser frei auf dem

ganzen Bildschirm positioniert werden. Die Ausmaße des Menüs passen sich dabei automatisch an. Es macht zwar keinen Sinn, Menüeinträge quer auf dem Bildschirm zu verteilen, aber wem es gefällt, kann es tun. Allerdings besteht dadurch auch die Möglichkeit, Menüeinträge nebeneinander zu positonieren. Einen Nachteil hat die Sache. Um in Menüeinträge neue Texte zu stellen, müssen diese mit der Maus angeklickt werden, wobei sie sehr leicht ungewollt verschoben werden. Aber auch für dieses Problem gibt es eine Lösung. Der Justiermodus für Menüs verhindert ein ungewolltes Verschieben und setzt alle Menüeinträge exakt untereinander.

Texte und Fonts

Alle Texte, seien es Menüeinträge, Gadget- oder Requestertexte, können in verschiedenen Schriftfonts angezeigt werden. Dazu erscheint ein Requester, in dem ein neuer Text eingegeben, Textfarbe und Zeichenmodus gewählt und der Zeichensatz und seine Attribute bestimmt werden können.

Wenden wir uns nun dem Ergebnis der Arbeit, dem eigentlich Wichtigsten des Programmes zu.

Das R.C.T. erzeugt aus den konstruierten Requestern und Menüleisten vier Dateien, wovon zwei für die weitere Bearbeitung wichtig sind. Das sind die Dateien mit der Endung ".RCT" und ".H".

Wie die Dateien weiter verwendet werden, hängt ganz von der Programmiersprache ab, die man einsetzen will. Die meisten Möglichkeiten haben die Programmierer, die in GFA-BASIC, C oder Assembler programmieren. Für diese Sprachen kann zusätzlich zu der Verwaltung der Dateien mit den Library-Funktionen noch der Source-Code erzeugt werden der sich direkt in die Programme miteinbinden läßt.

Leider kann man beim Abspeichern von R.C.T.-Dateien den Source-Code nicht direkt ausgeben lassen. Hierfür muß das Programm *"RCT_TO_ASCII"* gestartet werden. Nach dem Start öffnet sich auf der Workbench ein Filerequester, wie er

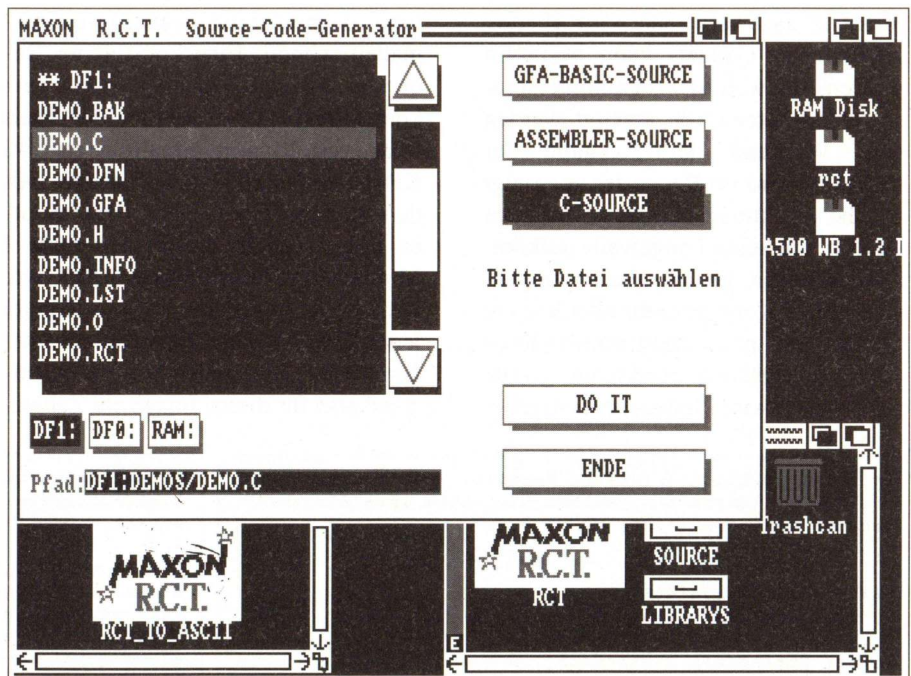


Abbildung 4: Mit dem Source-Code Generator können C-Assembler und C-Sources erzeugt werden. Leichter geht es kaum.

Abbildung 5: Dieser Requester mußte für den nachfolgenden Source Code herhalten. Die Eingabemöglichkeit wurde durch ein Stringgadget realisiert. Zur Eingabebestätigung dienen zwei Boolean-Gadgets.

```
#include <kick.h>
#include <exec/types.h>
#include <intuition/intuition.h>
#include <exec/memory.h>

struct TextAttr kicktxtattr[] = {
/* 0 */{(UBYTE *)"topaz.font",8,0,65},
/* 1 */{(UBYTE *)"topaz.font",8,0,65},
/* 2 */{(UBYTE *)"topaz.font",8,0,65},
/* 3 */{(UBYTE *)"topaz.font",9,2,73},
  };

struct IntuiText kickinttxt[] = {
/* 0 */{1,0,2,38,6,(struct TextAttr *)&kicktxtattr[0],(UBYTE
*)"Fertig",(struct IntuiText *)0L},
/* 1 */{1,0,2,8,-11,(struct TextAttr *)&kicktxtattr[1],(UBYTE *)"",
(struct IntuiText *)0L},
/* 2 */{1,0,2,26,6,(struct TextAttr *)&kicktxtattr[2],(UBYTE
*)"Abbrechen",(struct IntuiText *)0L},
```

im R.C.T. auch verwendet wird. Mit drei zusätzlichen Gadgets kann bestimmt werden, für welche Programmiersprache der Source-Code erzeugt werden soll. Während der Ausgabe der Datei(en) wird im Requester angezeigt welche Strukturen gerade geschrieben werden, um keine Langeweile aufkommen zu lassen. Bei einigen Umsetzungen geht die Konvertierung allerdings so schnell, daß mam nicht einmal lesen kann, an welchen Strukturen gerade geschrieben wird. Selbst die Konvertierung von Images in C-Source ist einigermaßen schnell. Eine konstuierter Requester mit einem Gadget, in das eine komplette IFF-Grafik (320 * 200 * 4) als Image gestellt wurde, ergab eine etwa 40 KByte umfassende R.C.T.-Datei. Mit dem Source-Code-Generator in C-Source umgewandelt, ergibt sich nach 2 Minuten eine Datei mit ca. 100 KByte Größe. Dies liegt daran, daß pro 16 Pixel jeder Bitplane der IFF-Grafik ein Integer (16 Bit) abgespeichert wird. Es entstand also für dieses Image ein Zahlenwust von 16000 Zahlen im ASCII-Format.

Bei solchen Spielereien sollte man darauf achten, daß die Diskettenkapazität ausreicht, da sonst die Konvertierung abgebrochen werden muß.

Zur Veranschaulichung, wie der erzeugte Source aussieht, hier zwei Beispiele. In der Abbildung 5 sehen Sie den Requester, dessen Source-Code untenstehend abgedruckt ist.

```
/* 3 */{1,0,2,25,14,(struct TextAttr *)&kicktxtattr
      [3],(UBYTE *)  "Machen Sie bitte Ihre Eingabe!",
(struct IntuiText *)0L},
  };

UBYTE strinfstraa[40];
UBYTE undostrinfstraa[40];

struct StringInfo kickstrinf[] = {
/* 0 */{(UBYTE *)&strinfstraa, (UBYTE
*)&undostrinfstraa, 0, 40, 0, 0, 0,
   0, 0, 0L, 0L, 0L},
  };

SHORT kickbrddat[] = {
/* 0 */ -1 , -1 ,124 , -1 ,124 , 21 , -1 , 21 , -1 ,-1
,
/* 10 */ -6 ,   3 ,  6 , -5 ,117 , -5 ,129 ,  3 ,129 ,
17 ,117 , 25 ,
    6 , 25 , -6 , 17 , -6 ,  3 ,
/* 28 */ -1 , -1 ,296 , -1 ,296 ,  8 , -1 ,  8 , -1 ,
-1 ,
/* 38 */ -1 , -1 ,124 , -1 ,124 , 21 , -1 , 21 , -1 , -
1 ,
/* 48 */ -6 ,   3 ,  6 , -5 ,117 , -5 ,129 ,  3 ,129 ,
17 ,117 , 25 ,
    6 , 25 , -6 , 17 , -6 ,  3 ,
/* 66 */  0 ,   0 ,355 ,  0 ,355 , 96 ,  0 , 96 ,  0 ,
0 ,
  };

struct Border kickbrd[] = {
/* 0 */{0, 0, 0, 0, 0, 5,(SHORT
*)&kickbrddat[0],(structBorder*)&kickbrd     [1]},
/* 1 */{0, 0, 3, 0, 0, 9,(SHORT
*)&kickbrddat[10],(struct Border *)0L},
/* 2 */{0, 0, 1, 0, 0, 5,(SHORT
*)&kickbrddat[28],(struct Border *)0L},
/* 3 */{0, 0, 0, 0, 0, 5,(SHORT
*)&kickbrddat[38],(struct Border *)&kickbrd    [4]},
/* 4 */{0, 0, 3, 0, 0, 9,(SHORT *)&kickbrddat[48],
(struct Border *)0L},
/* 5 */{4, 2, 0, 0, 0, 5,(SHORT *)&kickbrddat[66],
(struct Border *)0L},
  };

struct Gadget kickgad[] = {
/* 0 */{(struct Gadget *)&kickgad[1], 203, 64, 124, 21,
                                       0, 3, 4097,
       (struct Border *)&kickbrd[0],
       (LONG)0, (struct IntuiText *)&kickinttxt[0],
       (LONG)0, (LONG)0, 0, (LONG)0},
/* 1 */{(struct Gadget *)&kickgad[2], 35, 39, 296, 8,
                                      0, 3, 4100,
       (struct Border *)&kickbrd[2],
       (LONG)0, (struct IntuiText *)&kickinttxt[1],
       (LONG)0, (struct StringInfo *)&kickstrinf[0],1,
       (LONG)0},
/* 2 */{(struct Gadget *)0L, 32, 65, 124, 21, 0, 3,
                        4097,
       (struct Border *)&kickbrd[3], (LONG)0,
       (struct IntuiText *)&kickinttxt[2], (LONG)0,
       (LONG)0, 2, (LONG)0},
  };
```

```
struct Requester kickreq;
kickreqstart(window)
struct Window *window;
{
InitRequester(&kickreq);
kickreq.LeftEdge = 0;
kickreq.TopEdge = 0;
kickreq.Width = 364;
kickreq.Height = 101;
kickreq.BackFill = 2;
kickreq.Flags = 0;
kickreq.ReqGadget = &kickgad[0];
kickreq.ReqBorder = &kickbrd[5];
kickreq.ReqText = &kickinttxt[3];
return(Request(&kickreq,window));
}
```

Der erzeugte C-Source-Code

Was zum lauffähigen Programm noch fehlt, ist die *main()*-Funktion und ein Fenster für den Requester. Außerdem müssen die Daten der Images noch aufbereitet werden, wenn man eine Speichererweiterung an seinem Rechner hat. Zu diesem Zweck wird zusätzlich die Funktion *Imagemem()* mitgeneriert. Mit ihr werden die Daten des Images in den "Chipspeicher" kopiert. Zurück erhält man die Adresse des Speicherbereichs, an der das Image steht.

```
struct NewWindow NW =
  {
  0,11,640,245,0,1,GADGETDOWN|GADGETUP,
  WINDOWDRAG|WINDOWDEPTH|WINDOWSIZING|ACTIVATE|
  GIMMEZEROZERO|NOCAREREFRESH,
  NULL,NULL,"
Objektauswahl",NULL,NULL,1,1,640,245,WBENCHSCREEN
  };

struct IntuitionBase *IntuitionBase;
struct DosBase *DosBase;

main()
{
struct Window *W0;

if(!(IntuitionBase = (struct IntuitionBase
*)OpenLibrary("intuition.library",0)))
  exit(0);

if(!(DosBase = (struct DosBase
*)OpenLibrary("dos.library",0)))
  exit(0);

if(W0 = (struct Window *) OpenWindow(&NW))
  {
  kickimage[0].ImageData = (USHORT
*)Imagemem(&kickimgbitblk[0],48);
  kickimage[1].ImageData = (USHORT
*)Imagemem(&kickimgbitblk[48],48);
```

Holen Sie sich auch den neuen AMIGA-GRUNDLEHRGANG

DM 59,-

Buch und Diskette

unverbindlich empfohlener Verkaufspreis

WICHTIGE MERKMALE:

★ Das Buch für den richtigen Einstieg mit dem Commodore AMIGA ★ Auf über 400 Seiten werden dem Leser leicht verständlich die Grundlagen der Computertechnik und der Umgang mit Hardware erklärt ★ Ein ausführlicher Hauptteil ist dem Einsatz der grafischen Benutzeroberfläche des Betriebssystems gewidmet. Hier erläutert das Buch Fenster, Pulldown-Menüs und die vielen anderen Teile der ***Workbench*** ★ Wer die Maus nicht mag, der kann aus dem Kapitel über den ***Command Line Interpreter (CLI)*** entnehmen, wie man den AMIGA auch ohne Maus einsetzen kann ★ Ein weiterer Bereich des Buches ist die Einführung in die Programmiersprache ***BASIC***. Eine umfangreiche Befehlsübersicht sowie einige interessante Programme dienen der Erlernung und dem guten Training von BASIC ★ Anhänge wie z. B. ein ***Index*** und eine ***Sachworterklärung*** bieten das schnelle Nachschlagen und Auffinden wichtiger Punkte ★ ***Mit dem Buch erhalten Sie eine Programmdiskette mit allen abgedruckten Listings.*** Damit können die Beispielprogramme ohne die Mühe und Arbeit des Eintippens auf dem Computer nachvollzogen werden.

AUS DEM INHALT:

1. Die Hardware des AMIGA
★ die versch. AMIGA-Modelle ★ die Diskettenstation ★ Anschluß eines Druckers ★ Monitore am AMIGA ★ Erweiterung des AMIGA-Systems ★ Einstieg in die MS-DOS Welt mit dem AMIGA ★ Die „Innereien" des AMIGA (RAM, ROM u. Prozessoren)

2. Das Betriebssystem des AMIGA
★ Betriebssysteme und ihre Bedeutung ★ Die Benutzeroberfläche des AMIGA ★ Steuerung der Workbench ★ Arbeiten mit Maus, Fenstern und Pull-down-Menüs ★ Verwendung von Disketten, Dateien, Directory ★ Die Programme der Workbench Diskette im Einzelnen ★ Der CLI und seine Bedienung ★ Kopieren, Löschen und Batch-Bearbeitung im CLI

3. Programmieren in Amiga-Basic
★ Die Bedienung des Basic-Interpreters ★ Variable in Basic ★ Schleifenstrukturen ★ Die IF-Abfrage ★ Proceduren zur Programmstrukturierung ★ Graphik-Programmierung in AMIGA-BASIC ★ Dateiverwaltung ★ ausführliche Befehlsübersicht mit detaillierten Erklärungen

4. Zum Training
★ Programm-Diskette mit allen abgedruckten Listings ★ Sachworterklärung (Fachwörter-Lexikon) ★ Ausführlicher Index (Stichwortverzeichnis mit entspr. Verweisen)

BESTELL-COUPON an Heim-Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt

Ich bestelle ____________________
zzgl. DM 5,– Versandkosten (unabhängig von der bestellten Stückzahl)
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name, Vorname ____________________

Straße, Hausnummer ____________________

PLZ, Ort ____________________

Benutzen Sie auch die in KICKSTART vorhandene Bestellkarte

Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt-Eberstadt
Telefon 06151-56057

```
  if(kickreqstart(W0);
    {
    Delay(200);
    kickreqend();
    }
  CloseWindow(W0);
  if((kickimage[0].ImageData)
    FreeMem(kickimage[0].ImageData,96);
  if((kickimage[1].ImageData)
    FreeMem(kickimage[1].ImageData,96);
  }
CloseLibrary(IntuitionBase);
CloseLibrary(DosBase);
}
```

Der Header beinhaltet einige wichtige Funktionsaufrufe

Der gleiche Requester in der GFA-BASIC Version.

```
FUNCTION make_c_string(a$)
  LOCAL str%,c%
  IF LEN(a$)
    c%=MALLOC(LEN(a$),&H10000)
    IF c%
      str%=V:a$
      FOR i%=0 TO LEN(a$)
        BYTE{c%+i%}=BYTE{str%+i%}
      NEXT i%
      RETURN c%
    ENDIF
  ENDIF
  RETURN 0
ENDFUNC
PROCEDURE initrct
  REM Initialisierung der Textattribut Structuren
  txtattr%=MALLOC(32,&H10000)
  IF txtattr%
    LONG{txtattr%+0}=@make_c_string("topaz.font")!
Adresse des Fontname
    CARD{txtattr%+4}=8! ta_YSize
    BYTE{txtattr%+6}=0! ta_Style
    BYTE{txtattr%+7}=65! ta_Flags
    LONG{txtattr%+8}=@make_c_string("topaz.font")!
Adresse des Fontnames
    CARD{txtattr%+12}=8! ta_YSize
    BYTE{txtattr%+14}=0! ta_Style
    BYTE{txtattr%+15}=65! ta_Flags
    LONG{txtattr%+16}=@make_c_string("topaz.font")!
Adresse des Fontnames
    CARD{txtattr%+20}=8! ta_YSize
    BYTE{txtattr%+22}=0! ta_Style
    BYTE{txtattr%+23}=65! ta_Flags
    LONG{txtattr%+24}=@make_c_string("topaz.font")!
Adresse des Fontnames
    CARD{txtattr%+28}=9! ta_YSize
    BYTE{txtattr%+30}=2! ta_Style
    BYTE{txtattr%+31}=73! ta_Flags
  ENDIF
  REM Initialisierung der IntuiText Structuren
  itxt%=MALLOC(96,&H10000)
  IF itxt%
    BYTE{itxt%+0}=1! FrontPen
    BYTE{itxt%+1}=0! BackPen
    BYTE{itxt%+2}=2! DrawMode
    INT{itxt%+4}=38! LeftEdge
    INT{itxt%+6}=6! TopEdge
    LONG{itxt%+8}=ADD(txtattr%,0)! Adresse der TextAttr
                                   Structur
    LONG{itxt%+12}=@make_c_string("Fertig")! *IText
    LONG{itxt%+16}=0! NextText
    BYTE{itxt%+24}=1! FrontPen
    BYTE{itxt%+25}=0! BackPen
    BYTE{itxt%+26}=2! DrawMode
    INT{itxt%+28}=8! LeftEdge
    INT{itxt%+30}=-11! TopEdge
    LONG{itxt%+32}=ADD(txtattr%,8)! Adresse der
                                    TextAttr Structur
    LONG{itxt%+36}=@make_c_string("")! *IText
    LONG{itxt%+40}=0! NextText
    BYTE{itxt%+48}=1! FrontPen
    BYTE{itxt%+49}=0! BackPen
    BYTE{itxt%+50}=2! DrawMode
    INT{itxt%+52}=26! LeftEdge
    INT{itxt%+54}=6! TopEdge
    LONG{itxt%+56}=ADD(txtattr%,16)! Adresse der
                                     TextAttr Structur
    LONG{itxt%+60}=@make_c_string("Abbrechen")! *IText
    LONG{itxt%+64}=0! NextText
    BYTE{itxt%+72}=1! FrontPen
    BYTE{itxt%+73}=0! BackPen
    BYTE{itxt%+74}=2! DrawMode
    INT{itxt%+76}=25! LeftEdge
    INT{itxt%+78}=14! TopEdge
    LONG{itxt%+80}=ADD(txtattr%,24)! Adresse der
                                     TextAttr Structur
    LONG{itxt%+84}=@make_c_string("Machen Sie bitte
                            Ihre Eingabe!")! *IText
    LONG{itxt%+88}=0! NextText
  ENDIF
  REM Initialisierung der StringInfo Structuren
  strinf%=MALLOC(40,&H10000)
  IF strinf%
    LONG{strinf%+0}=@make_c_string(SPACE$(40))!Buffer
    LONG{strinf%+4}=@make_c_string(SPACE$(40))!UndoBuf.
    INT{strinf%+8}=0 ! BufferPos
    INT{strinf%+10}=40! MaxChars
    INT{strinf%+12}=0 ! DispPos
  ENDIF
  REM Initialisierung der Borderdaten
  brddat%=MALLOC(152,&H10000)
  IF brddat%
    borderdat:
    DATA  -1, -1, 124, -1, 124, 21, -1, 21, -1, -1, -6
    DATA   3, 6, -5, 117, -5, 129, 3, 129, 17, 117
    DATA  25, 6, 25, -6, 17, -6, 3, -1, -1, 296, -1
    DATA 296, 8, -1, 8, -1, -1, -1, -1, 124
    DATA  -1, 124, 21, -1, 21, -1, -1, -6, 3, 6, -5
    DATA 117, -5, 129, 3, 129, 17, 117, 25, 0
    DATA  25, -6, 17, -6, 3, 0, 0, 355, 0, 355, 96, 0
    DATA  96, 0, 0
    RESTORE borderdat
    FOR i%=0 TO 75
      READ x%
      INT{brddat%+i%*2}=x%
    NEXT i%
  ENDIF
  REM Initialisierung der Border Structuren
  border%=MALLOC(96,&H10000)
  IF border%
    INT{border%+0}=0! LeftEdge
    INT{border%+2}=0! TopEdge
    BYTE{border%+4}=0! FrontPen
    BYTE{border%+5}=0! BackPen
    BYTE{border%+6}=0! DrawMode
    BYTE{border%+7}=5! Count
    LONG{border%+8}=ADD(brddat%,0)! *XY
    LONG{border%+12}=ADD(border%,16)! NextBorder
    INT{border%+16}=0! LeftEdge
    INT{border%+18}=0! TopEdge
    BYTE{border%+20}=3! FrontPen
    BYTE{border%+21}=0! BackPen
    BYTE{border%+22}=0! DrawMode
    BYTE{border%+23}=9! Count
    LONG{border%+24}=ADD(brddat%,20)! *XY
    LONG{border%+28}=0! NextBorder
    INT{border%+32}=0! LeftEdge
    INT{border%+34}=0! TopEdge
    BYTE{border%+36}=1! FrontPen
    BYTE{border%+37}=0! BackPen
    BYTE{border%+38}=0! DrawMode
    BYTE{border%+39}=5! Count
    LONG{border%+40}=ADD(brddat%,56)! *XY
    LONG{border%+44}=0! NextBorder
    INT{border%+48}=0! LeftEdge
    INT{border%+50}=0! TopEdge
    BYTE{border%+52}=0! FrontPen
    BYTE{border%+53}=0! BackPen
    BYTE{border%+54}=0! DrawMode
    BYTE{border%+55}=5! Count
    LONG{border%+56}=ADD(brddat%,76)! *XY
    LONG{border%+60}=ADD(border%,64)! NextBorder
    INT{border%+64}=0! LeftEdge
    INT{border%+66}=0! TopEdge
```

```
    BYTE{border%+68}=3! FrontPen
    BYTE{border%+69}=0! BackPen
    BYTE{border%+70}=0! DrawMode
    BYTE{border%+71}=9! Count
    LONG{border%+72}=ADD(brddat%,96)! *XY
    LONG{border%+76}=0! NextBorder
    INT{border%+80}=4! LeftEdge
    INT{border%+82}=2! TopEdge
    BYTE{border%+84}=0! FrontPen
    BYTE{border%+85}=0! BackPen
    BYTE{border%+86}=0! DrawMode
    BYTE{border%+87}=5! Count
    LONG{border%+88}=ADD(brddat%,132)! *XY
    LONG{border%+92}=0! NextBorder
  ENDIF
  REM Initialisierung der Gadget Structuren
  gadget%=MALLOC(144,&H10000)
  IF gadget%
    LONG{gadget%+0}=ADD(gadget%,48)! NextGadget
    INT{gadget%+4}=203! LeftEdge
    INT{gadget%+6}=64! TopEdge
    INT{gadget%+8}=124! Width
    INT{gadget%+10}=21! Height
    CARD{gadget%+12}=0! Flags
    CARD{gadget%+14}=3! Activation
    CARD{gadget%+16}=4097! GadgetType
    LONG{gadget%+18}=border%+0! GadgetRender
    LONG{gadget%+26}=ADD(itxt%,0)! GadgetText
    CARD{gadget%+38}=0! GadgetID
    LONG{gadget%+48}=ADD(gadget%,96)! NextGadget
    INT{gadget%+52}=35! LeftEdge
    INT{gadget%+54}=39! TopEdge
    INT{gadget%+56}=296! Width
    INT{gadget%+58}=8! Height
    CARD{gadget%+60}=0! Flags
    CARD{gadget%+62}=3! Activation
    CARD{gadget%+64}=4100! GadgetType
    LONG{gadget%+82}=ADD(strinf%,0) ! StringInfo
    LONG{gadget%+66}=border%+32! GadgetRender
    LONG{gadget%+74}=ADD(itxt%,24)! GadgetText
    CARD{gadget%+86}=1! GadgetID
    LONG{gadget%+96}=0! NextGadget
    INT{gadget%+100}=32! LeftEdge
    INT{gadget%+102}=65! TopEdge
    INT{gadget%+104}=124! Width
    INT{gadget%+106}=21! Height
    CARD{gadget%+108}=0! Flags
    CARD{gadget%+110}=3! Activation
    CARD{gadget%+112}=4097! GadgetType
    LONG{gadget%+114}=border%+48! GadgetRender
    LONG{gadget%+122}=ADD(itxt%,48)! GadgetText
    CARD{gadget%+134}=2! GadgetID
  ENDIF
RETURN
PROCEDURE reqa(wind%)
  reqa%=MALLOC(120,&H10000)
  IF reqa%
    INT{reqa%+4}=0! LeftEdge
    INT{reqa%+6}=0! TopEdge
    INT{reqa%+8}=364! Width
    INT{reqa%+10}=101! Height
    LONG{reqa%+16}=ADD(gadget%,0)! ReqGadget
    LONG{reqa%+24}=ADD(itxt%,72)! ReqText
    LONG{reqa%+20}=ADD(border%,80)! ReqBorder
    CARD{reqa%+28}=0! Flags
    BYTE{reqa%+30}=2! BackFill
    ~Request(reqa%,wind%)
  ENDIF
RETURN
PROCEDURE reqaend
  IF reqa%
    ~EndRequest(reqa%,LONG{reqa%+72})
    ~MFREE(reqa%,120)
  ENDIF
RETURN
PROCEDURE rctfree
  ~MFREE(LONG{txtattr%+0},10)
  ~MFREE(LONG{itxt%+12},6)
  ~MFREE(LONG{txtattr%+8},10)
  ~MFREE(LONG{txtattr%+16},10)
  ~MFREE(LONG{itxt%+60},9)
  ~MFREE(LONG{txtattr%+24},10)
  ~MFREE(LONG{itxt%+84},31)
  ~MFREE(LONG{strinf%+0},40)
  ~MFREE(LONG{strinf%+4},40)
  IF txtattr%
    ~MFREE(txtattr%,32)
  ENDIF
  IF itxt%
    ~MFREE(itxt%,96)
  ENDIF
  IF strinf%
    ~MFREE(strinf%,40)
  ENDIF
  IF brddat%
    ~MFREE(brddat%,152)
  ENDIF
  IF border%
    ~MFREE(border%,96)
  ENDIF
  IF gadget%
    ~MFREE(gadget%,144)
  ENDIF
RETURN
```

Der erzeugte GFA-BASIC Source-Code

Der Aufwand, um die GFA-BASIC-Version starten zu können, ist zwar etwas geringer, man hat aber im Gegensatz zur C-Version ein ca. doppelt so langes Programm.

Das Startprogramm für die GFA-BASIC Version muß selbst geschrieben werden.

```
OPENW 1
FULLW 1
initrct
reqa(WINDOW(1))
ON BREAK GOSUB ende
DO
LOOP
'
PROCEDURE ende
  reqaend
  rctfree
  CLOSEW 1
  END
RETURN
```

Im C-Beispielprogramm muß die Struktur für das Fenster noch selbst eingegeben werden, während im BASIC ein Befehl genügt um das Fenster anzuzeigen. Damit es in C genauso einfach geht, können auch die Funktionen der R.C.T.-Library genutzt werden. Libraries sind Funktionsbibliotheken, die reentrant im Speicher stehen und somit von mehreren Programmen gleichzeitig genutzt werden können. Die Betriebssystemfunktion *OpenLibrary* dürfte keinem C-Pogrammierer unbekannt sein. Mit ihr werden die Libraries aus dem Libs-Ordner der Workbench-Diskette in den Speicher geladen und/oder die Basisadresse der Library ermittelt. Relativ zu der ermittelten Basisadresse werden die Funktionen der Library aufgerufen. Bevor die Funktion jedoch angesprungen werden kann, müssen ihre Parameter in die entsprechenden Register geladen werden, was mit einem C-Programm nicht machbar ist. Ein mitgeliefertes Assemblerprogramm erledigt diese Arbeit. Es muß lediglich assembliert und zum eigenen Programm gelinkt werden.

Die etwas kürzere *main()*-Funktion sieht dann wie folgt aus:

```
struct RctBase    /* Lib Base */
  {
  struct Library Library;
  BPTR   SegList;
  struct ExecBase *SysBase;
  struct IntuitionBase *IntuiTionBase;
  struct DosBase *DosBase;
  struct GfxBase *GfxBase;
  };

struct RctBase *RctBase; /* Lib Base Pointer */
struct DosBase *DosBase;

main()
{
APTR dummy;
struct Window *W0;

if(!(RctBase = (struct RctBase *)OpenLibrary("rct.library",0)))
  exit(0);

if(!(DosBase = (struct DosBase *)OpenLibrary("dos.library",0)))
  exit(0);

kickimage[0].ImageData = (USHORT *)Imagemem(&kickimgbitblk[0],48);
kickimage[1].ImageData = (USHORT *)Imagemem(&kickimgbitblk[48],48);

if(W0 = (struct Window *)reqdrawwindow(dummy,&kickreq,dummy,"DEMO");
  {
  Delay(200);
  reqend(dummy,&kickreq,W0);
  }
if((kickimage[0].ImageData)
  FreeMem(kickimage[0].ImageData,96);
if((kickimage[1].ImageData)
  FreeMem(kickimage[1].ImageData,96);

CloseLibrary(DosBase);
CloseLibrary(RctBase);
}
```

Aber nicht nur C-Sources können mit den Funktionen verwaltet werden. Die Funktion *rctload()* beispielweise lädt eine normale vom R.C.T. erzeugte Datei in den Speicher. Mit den 40 weiteren Funktionen der Library kann die geladene Datei verwaltet werden. Funktionen zum Anzeigen von Requestern und Menüleisten sind ebenso enthalten wie solche, mit denen Gadgets oder Menüeinträge während des Programmlaufes verändert werden können. Durch das Library-Konzept ist das R.C.T. für alle Programmiersprachen interessant, in denen Betriebssystemfunktionen aufgerufen werden können - und in welcher ist das nicht möglich?!

Vertrieb:

MAXON Computer
Industriestr. 26
6236 Eschborn
Tel: 06196/481811

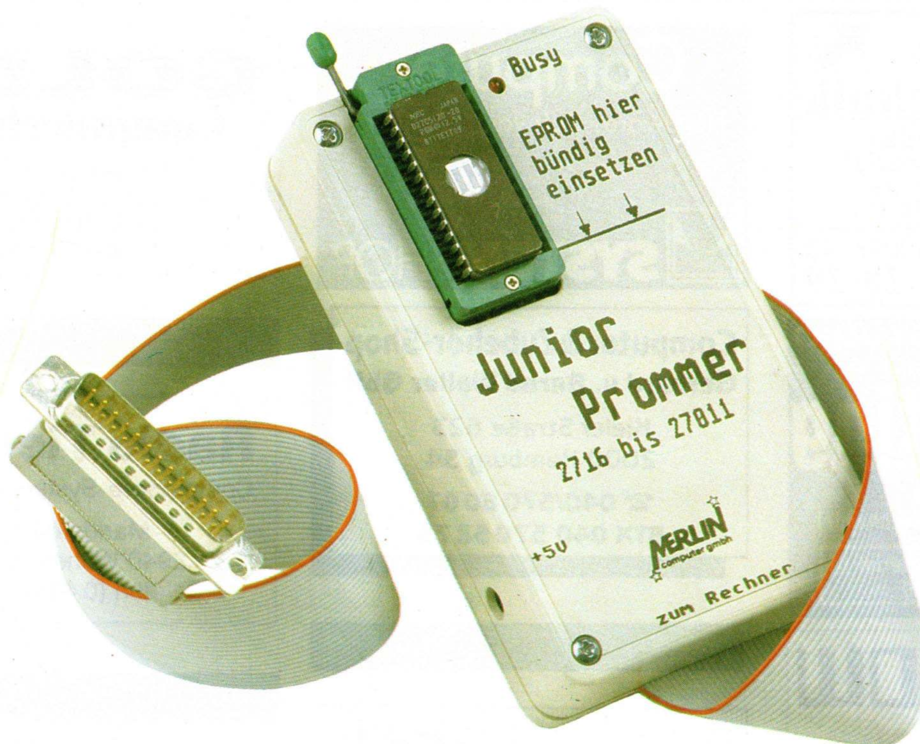
oder im Fachhandel

In diesem C-Header werden die R.C.T. -Library-Funktionen benutzt.

ENDE

Klein, kompakt und leistungsstark- der JuniorPrommer

HALLE 7 / E 42
HANNOVER MESSE
CeBIT '89
Welt-Centrum Büro · Information · Telekommunikation
8. - 15. MÄRZ 1989



Der JuniorPrommer programmiert alle gängigen EPROM-Typen, angefangen vom 2716 (2 KByte) bis zum modernen 27011 (1 MBit). Aber nicht nur EPROMS, sondern auch einige ROM- und EEPROM-Typen lassen sich lesen bzw. programmieren.

Zum Betrieb benötigt der JuniorPrommer nur +5V, die am Joystickport Ihres Amiga abgenommen werden; alle anderen Spannungen erzeugt die Elektronik des JuniorPrommers. Die sehr komfortable Software, natürlich Menü-unterstützt, erlaubt alle nur denkbaren-Manipulationen.

Fünf Programmieralgorithmen sorgen bei jedem EPROM-Typ für hohe Datensicherheit. Im eingebauten Hex-/ASCII-Monitor läßt sich der Inhalt eines EPROMS blitzschnell durchsuchen und auch ändern.

Alles dabei!

Bemerkenswert ist der Lieferumfang, so wird z.B. das Fertiggerät komplett aufgebaut und geprüft im Gehäuse mit allen Kabeln anschlußfertig geliefert.

Auf der Diskette mit der Treibersoftware befindet sich noch ein Programm, das die Kickstart-Diskette in vier EPROM-Dateien für 27512 (64 KByte EPROM) zerlegt, die dann mit dem JuniorPrommer gebrannt werden können. Ferner wird der Source-Code für die Lese- bzw. Programmierroutinen mitgeliefert. Und last but not least ist im Bedienungshandbuch (deutsch) der Schaltplan abgedruckt.

Bestellcoupon Merlin Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Hiermit bestelle ich:

☐ Fertiggerät wie oben beschrieben	DM 249,00
☐ Leerplatine (o. Bauteile) und Software	DM 59,00
☐ Leergehäuse (gebohrt und bedruckt)	DM 39,90

Versandkosten: Inland DM 7,50
Ausland DM 10,00
Auslandsbestellungen nur gegen Vorauskasse
Nachnahme zuzüglich DM 3,50 Nachnahmegebühr

Einkaufsführer

Hier finden Sie Ihren Commodore/Amiga Fachhändler

1000 Berlin

HD AMIGA Hardware Spezialist
Computertechnik
Sämtliche verfügbaren KICKSTART-PD's vorrätig! Und über 1500 weitere PD's für den AMIGA!
1000 Berlin 65 · Pankstr. 42
Tel. 030/465 70 28-29

COMPUTER-STUDIO
Schlichting
... die etwas andere Computerei
ATARI-Fachmarkt
MS-DOS Fachmarkt · NEC-Fachhandel
Katzbachstraße 8 · 1000 Berlin 61
☎ 030/7864340

RUNOW
Büroelektronik
Keithstraße 26 · 1000 Berlin 30
☎ 261126

SOFTPOWER
Das Software-Paradies im Norden Berlins!
Mo.-Fr. 11-18.30 * Sa. 10-14
Daily News!!!
IBM
1000 Berlin 65
Schwedenstr. 18c
Tel.: 030/492 20 56

2000 Hamburg

Bit Computer Shop
Osterstraße 173 · 2000 Hamburg 20
Telefon: 040/494400

Createam
Computer Hard & Software
Bramfelder Chaussee 300 · 2000 Hamburg 71
Telefon: 040/6416473+6416861

GMA mbH
040/2512416717
Systemhändler
Wandsbeker Chaussee 58
2000 Hamburg 76

2000 Hamburg

Computer
Hardware · Software · Zubehör
Lilienstraße 32
(beim Mönckebergbrunnen)
2000 Hamburg 1
Tel. (040) 336708
SYSTEMSHOP ®

Computer & Zubehör-Shop
Gerhard u. Bernd Waller GbR
Kieler Straße 623
2000 Hamburg 54
☎ 040/5706007
BTX 040 5705275

2160 Stade

BERGAU
EDV-Systeme
Büromaschinen · 2160 Stade
Neue Straße 5
Telefon (04141) 2364 u. 2384

2300 Kiel

Hardware
Software
Service
Spezialisiert auf Public Domain
Immer die neueste Software auf Lager
Home Computer Laden
Gutenbergstraße 5 · 2300 Kiel · Tel. (0431) 555555

2390 Flensburg

ecl electronic computer laden ohg
Norderstraße 94-96 · D-2390 Flensburg
☎ (0461) 28181 + 28193

2900 Oldenburg

GOLDT
Computerhaus
Donnerschweer Straße 127-129
(gegenüber Weser Ems Halle)
2900 Oldenburg
Telefon (0441) 884706

2940 Wilhelmshaven

Radio Tiemann
Commodore-Systemfachhändler
Marktstraße 52
2940 Wilhelmshaven
Telefon (04421) 26145

3000 Hannover

COM DATA
Am Schiffgraben 19 · 3000 Hannover 1
Telefon (0511) 326736

Fun Action
CSJ
Fun Action
Hannover's Softwarethek Nr. 1
DIE AUSWAHL an Software für C 16, C 64/128, Amiga, Atari ST u. IBM PC.
An der Tiefenriede 27 · 3000 Hannover 1
Tel. 0511/88 63 83

3500 Kassel

Hermann Fischer GmbH
Commodore-Systemfachhändler
Rudolf-Schwander-Str. 5-13
3500 Kassel
Telefon (0561) 700000

4500 Osnabrück

B.O.S.S. Computerhaus

Am Berliner Platz
Goethering 3
4500 Osnabrück
Telefon (05 41) 2 65 70

4650 Gelsenkirchen-Horst

MENTIS GmbH

Hard- und Software, Literatur
Bauteile, Service, Versand
Groß- und Einzelhandel

Poststraße 15 · 4650 Gelsenkirchen-Horst
Telefon (02 09) 5 25 72

6000 Frankfurt

Oederweg 7-9
6000 Frankfurt/Main 1
☎ (0 69) 55 04 56 - 57

Commodore OKI ATARI TOSHIBA

6200 Wiesbaden

DTM
Luisenstraße 47
6200 Wiesbaden
Telefon (0 61 21) 56 20 91
Computershop GmbH

AUTORISIERTER
COMMODORE
SYSTEM-HÄNDLER

COMPUTER TREFF

Computerbedarf, PD und Software für ATARI, AMIGA, PC

Nettelbeckstraße 12
6200 Wiesbaden
Tel. (0 61 21) 40 43 02

6380 Bad-Homburg

PDC GmbH
Produkte u. Details Computerverband

Louisenstraße 115
Ladenpassage Alter Bahnhof
6380 Bad-Homburg
Telefon (0 61 72) 2 47 48

Autorisierter Commodore-Systemhändler!

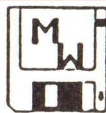
6457 Maintal

Landolt-Computer
Beratung · Service · Verkauf · Leasing

Autorisierter Commodore-Händler

Wingertstr. 112 · 6457 Maintal/Dörnigheim
Telefon (0 61 81) 4 52 93

6551 Fürfeld



MICHAEL WEISGERBER

HARD + SOFT

Rathausstraße 2
6551 Fürfeld
Telefon (0 67 09) 7 78

6680 Neunkirchen

SHOP 64
Computer GmbH
Neunkirchen
Telefon (0 68 21) 2 37 13

Commodore Systemhändler

Saarbrücken
Saarlouis
Homburg
St. Ingbert
Neustadt
Trier
Kaiserslautern

6800 Mannheim

Computersysteme + Textsysteme

6800 Mannheim 24
Casterfeldstraße 74-76
Telefon (06 21) 85 00 40 · Teletex 6 211 912

7000 Stuttgart

»If AMIGA, go to Schreiber«
Stuttgart's starker Computer-Laden.
SCHREIBER COMPUTER
Rotebühlplatz 10
7000 Stuttgart-1
Tel. 0711/227099
Neu · Neu · Neu · Neu
Im SUBWAY
BREUNINGER City

7800 Freiburg

Comp.Z.

Pochgasse 31
7800 Freiburg
T.0761/554280

AUTORISIERTER
COMMODORE
SYSTEM-HÄNDLER

7890 Waldshut-Tiengen

hettler-data
service gmbh

Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon (0 77 51) 30 94

8000 München

Commodore

AMIGA + PC, HARD- UND SOFTWARE
— *BERATUNG / VORFÜHRUNG* —
VIDEOBEARBEITUNG

MODL
FOTO · VIDEO · AUDIO · FERNSEH
COMPUTER · GELEGENHEITSMARKT
REPARATUREN · FOTOKOPIEN
FARBKOPIEN · PASSBILDER
MAX-WEBER-PLATZ · 8000 MÜNCHEN 80 · ☎ 4 80 16 50

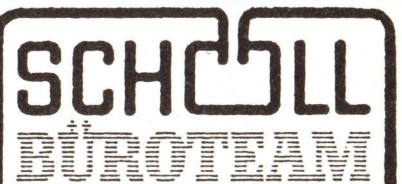
8400 Regensburg

Zimmermann elektroland

8400 Regensburg
Dr.-Gessler-Str. 8
☎ 09 41 / 9 50 85

8390 Passau
Kohlbruck 2a
☎ 08 51 / 5 20 07

8700 Würzburg



Hardware · Software
Service · Schulung

computer center
am Dominikanerplatz
Ruf (09 31) 3 08 08 - 0

Schweiz

Ihr Computer Spezialist

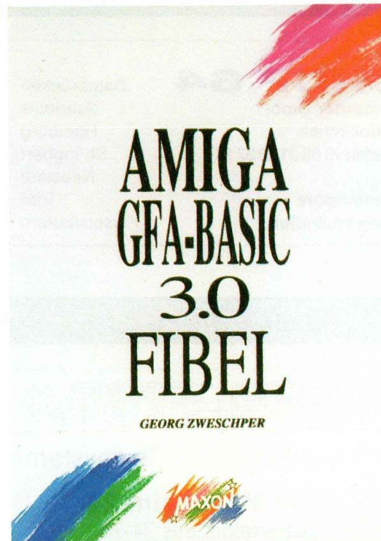
5000 Aarau, Bahnhofstrasse 86, Tel. 064/22 78 40
4102 Basel-Binningen, Kronenplatz, Tel. 061/47 88 64
5430 Wettingen, Zentralstrasse 93, Tel. 056/27 16 60
8400 Winterthur, St. Gallerstrasse 41, Tel. 052/27 96 96
8021 Zürich, Langstrasse 31, Tel. 01/241 73 73

Grösste Auswahl an Peripherie, Software, Literatur und Zubehör.

ATARI
Commodore

AMIGA
GFA-BASIC 3.0 Fibel

HALLE 7 / E 42
HANNOVER MESSE
CeBIT'89
Welt-Centrum Büro · Information · Telekommunikation
8. – 15. MÄRZ 1989



NEU!

Alles

Editor, Variablentypen, Befehle, Funktionen und Operatoren sind vollständig aufgeführt und beschrieben. Tastaturbelegung, Füllmuster und Fehlermeldungen sowie eine Liste aller implementierten Betriebssystem-Funktionen des AMIGA befinden sich im Anhang.
Dieses Buch eignet sich deshalb für alle Programmierer, die in diesen neuen und einzigartigen BASIC-Dialekt einsteigen möchten.

Ausführlich

Zu jedem Befehl wird eine genaue Beschreibung der Syntax und der Parameter gegeben. Die detaillierten, aber kompakten Erklärungen lassen keine Unklarheiten bestehen. Zusätzliche Informationen und Hinweise machen die Lektüre komplett und das Verstehen einfach.
Die Programmbeispiele zeigen den richtigen Umgang mit dieser mächtigen Programmiersprache, für den BASIC-Einsteiger wie für den AMIGA-Insider. Das Buch für jeden GFA-BASIC-Programmierer.

Alphabetisch

Wenn Sie mit den fast 400 Befehlen und Funktionen von GFA-BASIC arbeiten und die Syntax oder Bedeutung einzelner Befehle nachschlagen möchten, dann werden Sie diese Vorzüge, die Ihnen eine alphabetische Sortierung bietet, nicht mehr missen wollen.
Befehle, Funktionen, Operatoren und Systemvariablen sind deshalb streng alphabetisch geordnet und vereinfachen das Auffinden der gewünschten Informationen erheblich, denn Sie benötigen kein Inhaltsverzeichnis und keinen Index.

Bestellcoupon MAXON Computer GmbH Industriestraße 26 6236 Eschborn Tel.: 06196/481811

Name: ______________________
Vorname: ______________________
Straße: ______________________
Ort: ______________________
Unterschrift: ______________________

Hiermit bestelle ich:

☐ Exemplar(e) "AMIGA GFA-BASIC 3.0 Fibel" DM 39,00

Versandkosten: DM 7,50
Nachnahme zuzgl. DM 3,50 Nachnahmegebühr.

☐ Vorauskasse
☐ Nachnahme

Vertrieb in der Schweiz: DTZ DataTrade AG Langstrasse 94 Postfach 413 CH-8021 Zürich Tel.: 01/242 80 88 Fax.: 01/291 05 07
Vertrieb in Österreich: Dipl.-Ing. Reinhart Temmel Ges.m.b.H. & Co.KG Markt 109 A-5440 Golling Tel.: 06244/7081-17 Fax.: 06244/7188-3
Vertrtieb in Holland: JOTKA COMPUTING Postbus 8183 NL-6710 AD Ede Tel.: 08380/38731 Fax.: 08380/21675

LESERFORUM

Briefe an KICKSTART

Sehr geehrte Damen und Herren,

beiliegend sende ich Ihnen Kopien meiner Korrespondenz mit einem bekannten Fachbuchverlag. Leider mußte ich schon mehrfach feststellen, daß die gesamte Computerbranche, bzw. fast die ganze, nicht gerade kundenfreundlich ist. Das fängt damit an, daß, zumindestens in unserer weitesten Umgebung, in den Computershops nicht gerade geschultes Personal ist, daß man bald für jedes Anwenderprogramm ein weiteres benötigt, daß die Handbücher zu Programmen für Programmierer geschrieben werden und nicht für den "Otto-Normalverbraucher" und dann noch meist in Englisch; die Bücher, die dann schlau machen sollen, sind umständlich zu handhaben und sehr gewöhnungsbedürftig. Es gibt auch Computerzeitschriften, die es nicht für nötig befinden, Briefe zu beantworten.

Und jetzt dazu die Post mit dem Fachbuchverlag. Immmer wieder wird man um seine Meinung gefragt, soll sich mit Fragen an die Zeitschriften wenden und heraus kommt 0 (Null). Da braucht sich niemand mehr zu wundern, wenn der Computer nur zum Spielen benutzt wird und die teuren Anwenderprogramme nicht verkauft werden. Ich z.B. ärgere mich und verliere schon fast die Lust, programmieren zu lernen.

Gott sei Dank gibt es noch die PD-Programme, wenn auch hier leider meistens in Englisch.

Mit freundlichen Grüßen
B. Stenglin, Huntlosen

Sehr geehrte Damen und Herren,

Ihren Artikel "Zart getastet" habe ich mit Interesse gelesen.
Ich besitze einen AMIGA 2000, und bei mir ergibt sich jetzt die Frage, ob man mit Hilfe eines Scanners Börsenkurse aus einer Tageszeitung (laut Muster) in den Computer einlesen und weiterverarbeiten kann. Dabei müßten die Aktien und Kurse aus der Grafik herausgelöst und als String auf Diskette abgespeichert werden.

F. Janning, Recke

Sehr geehrter Herr Janning,
grundsätzlich ist dies möglich, es sprechen aber zwei Gründe dagegen:
1. Scanner mit der nötigen Qualität sind sehr teuer.
2. Passende Software ist für den AMIGA nicht erhältlich.
Der finanzielle Aufwand für ein speziell geschriebenes Programm ist auch nicht unerheblich, so daß die Lösung praktisch uninteressant ist.

Die Red.

Sehr geehrte Redaktion,

laut Sidecar-Handbuch ist es möglich, AMIGA-Diskettenlaufwerke auch dann vom PC aus anzusprechen, wenn diese NICHT am PC-Teil angeschlossen sind. Oder umgekehrt!

Doch wie?

Ich hoffe, Ihr wißt Rat für mich und all die anderen Besitzer eines Sidecars.

Mit freundlichen Grüßen
Peter ???

Lieber Peter,
die im Sidecarhandbuch beschriebene Zugriffsmöglichkeit besteht nur auf Dateiebene. PC-Disketten können nicht im AMIGA vom PC gelesen werden. Einzelne Dateien kann man aber übertragen. Ebenfalls ist es möglich, auf einer AMIGA-Diskette eine Datei zu öffnen, die vom PC wie eine Diskette verwaltet wird. Die Handhabung dieser Befehle wurde ausführlich im Heft 12/87 erläutert, das Sie noch bei uns nachbestellen können.

Die Red.

Sehr geehrte Damen und Herren,

ich bin erst seit kurzem stolzer Besitzer eines AMIGA 2000, aber bereits jetzt empfinde ich den relativ lauten Föhn (Ventilator) als nervend. Wie bekomme ich ihn leise, bzw. gibt es einen "Flüster-Föhn" zu kaufen?

Mit freundlichen Grüßen
Ihr entnervter Leser
S. Klabunde, Berlin

Zwei Lösungen gibt es für Ihr Problem:
1. einen Vorwiderstand in die Zuleitung des Lüfters schalten, was die Drehzahl und den Geräuschpegel absenkt oder 2. einen temperaturabhängigen Schalter (aus dem Bastelladen) in die Versorgung des Lüfters klemmen. Dadurch lärmt der "Föhn" wenigstens nur, wenn es nötig ist.

Die Red.

Sehr geehrte Damen und Herren,

hiermit möchte ich meiner Enttäuschung Ausdruck geben, da ich mich von Ihnen ver"......" fühle.

Voller Begeisterung las ich in der Maiausgabe '88 den Artikel über den KICKSTARTER. Eine tolle Möglichkeit, meinen A 2000 schneller zu machen und ihm ein besseres Bild zu verpassen - dachte ich. Da Sie mehrmals darauf hingewiesen haben, daß der KICKSTARTER nur mit einem Multisync-Monitor läuft, habe ich angenommen, daß er auch auf meinem NEC JC-1401 P3ED (ist der etwa kein Multisync-Monitor?) ein Superbild bringt. Vielleicht sollte ich das Denken den Pferden....? Aber alles, was ich zu sehen bekam war ein schreckliches Geflimmere. Ich muß Ihnen sagen, ich war und bin noch schwer enttäuscht. Jetzt weiß ich, daß mein Monitor nur in der Lage ist, 62Hz vertikal zu synchronisieren. So habe ich DM 60,- Material in den Wind geschrieben und die Schaltung zur Abschreckung an die Wand gehängt.

Damit es anderen Multisync-Besitzern nicht ähnlich frustrierend ergeht, möchte ich Sie doch bitten, die Leser zu informieren oder meinen Brief in Kurzform zu veröffentlichen. Ihre Leserschaft wird es Ihnen danken.

Vielleicht läuft der KICKSTARTER aber auch mit einem 34 Mhz Quarz-Oszillator? Falls ja, und falls es einen solchen OSC überhaupt gibt, würde ich eine Antwort von Ihnen zu schätzen wissen.

Trotzdem besten Dank.
St. Tünnermann

Sehr geehrter Herr Tünnermann,
Ihre Probleme bezüglich der Synchronisation sind uns leider unverständlich. Nach Durchsicht von Zuschriften und Nachfragen in unserem Einzugsbereich mußten wir feststellen, daß Sie der einzige von etlichen Multisync-Besitzern sind, der irgendwelche Probleme dieser Art hat. Auch Anfragen bei Technikern ergaben nichts Negatives diesbezüglich. Könnte es vielleicht an Toleranzen Ihres Monitors liegen? Ein 34 MHz Quarz-Oszillator funktioniert auch, über die Verfügbarkeit können wir leider nichts sagen.
Die Red.

Sehr gehrte Damen und Herren,

zunächst einmal möchte ich mich für eine hervorragende Computerzeitschrift bedanken.

In der Ausgabe 11/88 las ich mit Interesse den Vergleich der beiden 24-Nadeldrucker. Der Beitrag war gut gemacht, und ich stimme in großen Zügen mit Ihnen überein, was die Qualitäten der Drucker betrifft. Nur weniges haben Sie nicht erwähnt -meiner Meinung nach hätte unbedingt eine Bewertung in den Artikel kommen sollen, wie haltbar bzw. "auslutschbar" die Farbbänder der Kontrahenten sind. Dauertests gehören zu den wenigen Prüfungen, die sich dem gewöhnlichen Käufer entziehen.

Der zweite Anlaß meines Briefes bezieht sich auf Kickup - die mehrteilige Tragödie eines Spieleprogrammierers. Sein erfrischender Schreibstil bleibe ihm unbenommen! Aber während unsereiner sich längst damit abgefunden hat, daß Multitasking sich mit Spielen einfach nicht verträgt, sehe ich mit Grauen, WIE das Multitasking abgeschaltet wird. Hier wurde mit allen Regeln der Kunst gezeigt, wie man es NICHT machen sollte.

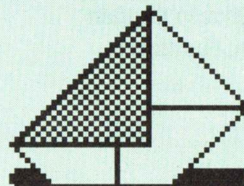
Angefangen von den fest eincodierten Customchipadressen (Es wird nie einen anderen AMIGA geben? Zumindest keinen, auf dem unsere Software läuft!), bis hin zu No-Nos mit dem Supervisormodus. Da türkt der Mann mit Exception-Vektoren herum, wo das Betriebssystem doch einen Befehl namens *SuperState()* offeriert.

Was macht den Leuten solche Schwierigkeiten, die Maschine auf dem richtigen Weg zu übernehmen? Sie hat für alles ihr Resourcen, und wenn man den Blitter mit *OwnBlit()* übernimmt, bekommt man schließlich auch dessen Adresse mitgeliefert. Wenn ich dann noch an das Umbiegen der Interrupts denke - ich hätte den Artikel wirklich nicht lesen sollen. Nur aus anderen Gründen, als der Autor vermutet.

Mit freundlichen Grüßen
P. Vohmann, Durlach

Sehr geehrter Herr Durlach,
mein Baßlehrer würde Ihnen wohl sagen: "Man kann sich auch ein Klavier um den Bauch binden". Haben Sie schon mal versucht, ein (Action-)Spiel unter dem Betriebssystem zu schreiben? Ich schon. Ich bin fast verzweifelt dabei. Um z.B. Hardwarescrolling zu machen, braucht die graphics.library fast einen ganzen Frame. Wann soll denn da noch das eigentliche Spiel laufen?
Ich kenne das AMIGA-BS jetzt seit der Kickstart 1.0 (seit etwa drei Jahren). Und seit der 1.0 KS hat es immer dieselben drei Bugs: Es stürzt ohne Grund ab, und es ist langsam oder unflexibel. Da ich aber die Hintergründe kenne, schimpfe ich nicht 'drüber - ich verwende einfach meine eigenen Routinen.
Ich weiß übrigens kein professionelles Spiel, das nach Ihren Vorstellungen arbeitet, aber vielleicht gelingt es ja Ihnen.
Martin der Silbernagl

Leserbriefe geben nicht unbedingt die Meinung der Red. wieder. Die Red. behält sich vor, Zuschriften sinngemäß zu kürzen.



DAS NEUESTE VOM SPIELEMARKT

cpl / ddf

Transputor

Derek Baker, Jon Low und Clement Chambers spekulieren anscheinend darauf, daß Ihnen die Ähnlichkeit mit "TRANSPUTER" Kunden bringen wird. Sie haben es nötig. TRANSPUTOR ist ein Break-Out-Spiel - das einhundertste. Zur Feier dieses Jubiläums spielt sich das Ganze dreidimensional ab. Das einzige Ergebnis des 3D-Effektes ist allerdings, daß das Spiel unübersichtlich wird. Das Paddle läßt sich wahlweise mit den Cursortasten oder dem Joystick hin- und herschieben. In beiden Fällen ist die Steuerung gleich miserabel. Aber wie schon Birne so schön sagt: "...alles ist eine Sache der Perspektive und man muß einfach nur positiv denken..." Positiv ist an Transputor, daß man alle 32 Levels ansehen und direkt anwählen kann. Ob dies allerdings genügt, um TRANSPUTOR zu einem gelungenen Programm zu machen, wage ich zu bezweifeln.

Tracksuit-Manager

Es ist Juli. Ihr Land ist in der Fußballweltmeisterschaft mit einer miserablen Leistung vorzeitig ausgeschieden. Als neuer Manager des Nationalteams haben Sie zwei Jahre Zeit, eine Mannschaft aufzubauen und die Qualifikation für den nächsten Nationen-Cup zu schaffen. Soviel zum Spielhintergrund. Tracksuit-Manager ist ein reines Strategiespiel, bei dem es darum geht, ein Team richtig auszuwählen, aufzustellen, für jeden Gegner die Spieltaktik auszuklügeln u.a.m..

Im Hauptmenü gibt es die Optionen: Date, Squad, Scout, Play, Watch, Cup, Diary und Save. Sie beinhalten bereits alles, was Sie als Team-Manager tun können.

Squad: Zuerst wird unter den besten Spielern des Landes ein Team von 15 bis 22 Spielern ausgewählt - die erweiterte Nationalmannschaft.

Diary: Hier werden die Freundschaftsspiele und Rundreisen des Teams arrangiert. Um die Mannschaft in Form zu bringen, empfiehlt es sich, sie zwischen den Qualifikationsspielen ab und zu durch ein Freundschaftsspiel in Form zu halten.

Scout: Der für das Team tätige Beobachter sieht sich die Gegner an. Seine Auskünfte über Spielstärke und Taktik der anderen Mannschaften sind wichtig für die eigene Strategie.

Watch: Wenn Sie selbst ein Spiel "beobachten" wollen, brauchen Sie keinen Scout zu schicken. Zu "sehen" gibt es allerdings nicht viel, denn die kompletten 90 Minuten laufen als Textausgabe über den Bildschirm, beispielsweise: "...Lineker plays a short pass to Baker, offside ...". Dieser Teil des Programmes ist durch seine Langatmigkeit eher ermüdend und sollte tunlichst gemieden werden.

Cup: Der aktuelle Stand der Qualifikation, die einzelnen Gruppen und ihre Tabellen sind in diesem Unterpunkt jederzeit abrufbereit.

Play: Bevor das Spiel per Textausgabe (siehe oben) über den Bildschirm geht, können Sie noch das Team und seine Taktik auswählen. Danach bleibt Ihnen nur Zuschauen und Warten aufs Ergebnis.

Save: Das Programm ist nicht kopiergeschützt. Sie sollten sich in jedem Fall zuerst ein Backup machen und nur damit spielen. Uns selbst ist es beim Speichern zweimal passiert, daß wir hinterher einen READ-WRITE-ERROR auf der Disk hatten.

Insgesamt ist Tracksuit-Manager ein Spiel für Strategen. Manche Spielteile sind etwas zu langatmig geraten. Dem Fußballfan wird auf dem AMIGA sowieso allerhand zugemutet - unseren mittlerweile bescheidenen Ansprüchen wird das Programm jedoch gerecht.

Phantom Fighter

Trotz des Namens ist Phantom Fighter von Martech keine Tiefflugsimulation. Eines hat es allerdings mit den gleichnamigen Flugzeugen der Luftwaffe gemeinsam: Wir wären besser davon verschont geblieben. Im ersten Augenblick präsentiert sich hier ein weiteres Schießspiel des Strickmusters: Raumschiff - Aliens - Feuer frei! Das Titelbild und die Graphik sind ganz ausgezeichnet. Aber das war es dann auch. Die monotone Musik treibt einen spätestens nach fünf Minuten zum Wahnsinn. Die Steuerung ist miserabel, die Angriffsformationen der Gegner ruckeln, der Bildschirm flackert, und in das Ganze ist kein Plan zu bekommen. Irgendwie erscheint das Spiel wie eine Super-Graphik-Demo, in die nachträglich eine Kollisionsabfrage eingebaut wurde. Schade, dies hätte ein prima Spiel werden können, es zeigt gute Ansätze. Vielleicht sollte sich der Graphiker einen besseren Programmierer suchen.

The Munsters

It is fun time down at the goulish mansion - bereits die Titelmusik schafft gute Laune. Sie spielen den Part von Mrs. Munster. Die Aufgabe ist simpel - retten Sie Marylin. Das ist allerdings nicht so leicht. Gespenster und Poltergeister schweben durchs Geisterhaus, die nur mittels der richtigen magischen Waffen getötet werden können. Wenn es den bösen Geschöpfen gelingt, Mrs. Munster zu berühren, schwindet die Lebensenergie, und alle Munsters sind verloren. Taktik spielt eine große Rolle bei dieser vergnüglichen Gespensterhatz. Wenn Sie in einem Raum zu lange zögern, erscheinen mehr und mehr Geister, und die Situation wird richtig brenzlig. Das Spiel beginnt im Geisterhaus und Ihr erstes Ziel ist es, ins Freie zu gelangen, wo einige außerodentlich wirksame Zauberwaffen liegen. Die Ausgänge werden deshalb von besonders bösen Geistern bewacht. Außerhalb des Hauses befinden sich ein Friedhof, eine Kapelle und auch ein Chateau, in dem Marilyn vom "bösen Nick" gefangengehalten wird. Die witzige Aufmachung und zahlreiche Trickeffekte geben den MUNSTERS einen unterhaltsamen Rahmen. Hier ist AGAIN & AGAIN wirklich ein Spiel gelungen, das man wieder und wieder spielen kann.

Gauntlet II

Der gemeinsame Siegeszug von Elf, Walküre, Zauberer und Krieger durch die Unterwelt (Gauntlet) wurde bereits 1987 von Andreas Hommel mit dem Spiel Garrisson AMIGA-gerecht umgesetzt. US Gold präsentiert jetzt mit der AMIGA-Version Gauntlet II eine "Originalumsetzung" des gleichnamigen Spielhallenautomaten. Die Umsetzung ist gut gelungen, Graphik und Spielsteuerung sind ansprechend. Ob man allerdings mit dem in die Jahre gekommenen Thema noch allzuviele Käufer in die Softwareshops locken kann, bleibt abzuwarten.

Chrysalis Software Ltd.

Das Softwarehaus Teque, bekannt geworden durch Terramex und Pacmania hat seinen Namen geändert in Chrysalis. Mit dem neuen Label wird auch der Anspruch erhoben, in Zukunft anstelle von Konvertierungen lieber Spiele nach "eigenen Ideen" zu machen. Die ersten drei Titel werden bereits entwickelt: Advanced Ski - dessen Slalomstrecke ein wenig an Marble Madness erinnert, Advanced Rugby - der Urvater des "American Football" und Prison - ein Spiel, das ein wenig an Oblitegrator und Alien Legion erinnert.

UMS - Universal Military Simulation

Sehr vorsichtig behandelt die KICKSTART-Redaktion Spiele, die den Krieg zum Hauptthema gewählt haben. Nicht etwa wegen der jüngsten deutschen Geschichte, sondern weil es kaum etwas Perverseres gibt, als Menschen bewußt zu verletzen oder zu töten. Wir sind uns deshalb auch jetzt noch nicht sicher, wie wir UMS bewerten sollen. Einerseits ist es ein brillant gemachtes Programm, bei dem es im Gegensatz zu Veteran, POW und Platoon nicht darum geht, alles abzuballern, was einem vor die Flinte kommt; ja der Krieg findet sogar nur als strategisches Spiel in Form von zu verschiebenden Einheiten statt. Andererseits könnte dies gerade deshalb gefährlich sein, weil Krieg hier nun ganz ohne Schrecken, eben nur als taktisches Spektakel dargestellt wird. Wie auch immer, das Thema ist heikel, das Programm als solches ist hervorragend, wir sind dankbar, daß nicht wie bei manch anderem "unterhaltsamen Spielchen" das Blut nur so über den Bildschirm spritzt - und solange Sie sich bewußt bleiben, worin der Unterschied zwischen UMS und der Wirklichkeit liegt, sollen uns auch die zahlreichen Szenario-Disks nicht stören, die Rainbird mittlerweile dazu herausbringt.

Thunder Blade

Roy Scheider als Hubschrauberpilot im Film Blue Thunder gab den Anstoß zum Spielhallenautomaten THUNDER BLADE von Sega. Die Hintergrundstory ist so neu wie ein Faschingskrapfen am Aschermittwoch (General Swindells - böse - beschließt Invasion; Du - braver Superheld - besteigst Hubschrauber; peng!). Was den Automaten in den Arcades so erfolgreich gemacht hat, war die phantastische Graphik, verbunden mit gelungener Animation und äußerst realistischer Darstellung der Helikoptersteuerung. Die AMIGA-Version kommt dem Automatenoriginal recht nahe, trotzdem bleibt auch hier der schale Geschmack eines Spieles, bei dem das Abschießen zum Selbstzweck geworden ist. Weitere Neuerscheinungen mit gleichem Thema und ähnlichem Inhalt sind: Hellfire Attack (Martech), Afterburner (Activision), F-16 Combat Pilot (Digital Integration) und wohl im Herbst F-19 Stealth Fighter (Micro Prose).

Flying Shark

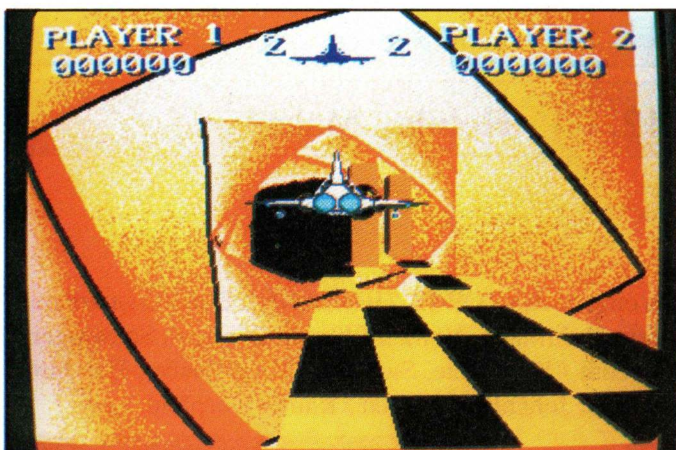
Eine weitere Automatenumsetzung präsentiert in diesen Tagen das englische Softwarehaus Firebird - FLYING SHARK von Taito. Der fliegende Hai tummelt sich allerdings nicht im Wasser, sondern in der Luft und auch dieses Spiel "...is a fast and furious vertical scrolling shoot-em-up...". Durch insgesamt fünf Levels schießt und bombt sich der Pilot, konfrontiert mit gegnerischen Flugzeuggeschwadern, Panzern, Flakgeschützen, Kriegsschiffen und Patrouillenbooten. Fertig wird Flying Shark im März; aber das Erscheinungsdatum spielt eigentlich keine Rolle, wenn man die Originalität des Programmes berücksichtigt. (Je später, desto besser).

Zak McKracken

Zak, der Held im neuesten Lucasfilm-Adventure, versucht, die Erde vor einem Anschlag der Marsbewohner zu bewahren. Zak ist Reporter beim "National Inquisitor" (Parodie auf Amerikas National Inquirer), einem Blatt, das die Bild-Zeitung als Lektüre für Kinder im Vorschulalter erscheinen läßt. Durch Schlagzeilen erhält er Hinweise, die er zur Rettung der Welt vor der vom Mars eingeschleppten Dummheitsseuche benötigt. Zur Steuerung des Helden gibt es ein Menü mit Verben. Diese können mit den gefundenen Gegenständen per Maus kombiniert werden (beispielsweise: give book to police officer). Das Spiel ist bisher nur in englischer Sprache erhältlich. Wenn Sie damit zurechtkommen und Abenteuer im Stil von "Leisure Suit Larry" (Teil II erscheint übrigens in Kürze) mögen, liegen Sie mit Zak McKracken richtig.

Slip Stream

Kurz vor Redaktionsschluß hat uns Microdeal noch ein paar Bilder von Slip Stream zukommen lassen. Die Graphik zeigt ein Flugzeug vor verschiedenen Hintergründen. Eine Story dazu hat man uns erspart. Uns wäre auch bestimmt nichts mehr eingefallen, um "noch ein Flugzeug fliegt durch noch eine Landschaft und kämpft mit noch mehr Feinden" zu beschreiben (siehe oben, weiter oben und ganz oben). Zugegeben, die Graphik macht einen guten Eindruck. Warten wir also ab und lassen uns überraschen, ob Slip Stream auch sonst etwas Neues bietet.



TOP 12

1.	(4)	Elite

2.	(1)	Interceptor
3.	(2)	Ports of Call
4.	(-)	Dungeon Master
5.	(3)	Starglider II
6.	(7)	Bards Tale II
7.	(5)	Carrier Command
8.	(-)	Dragons Lair
9.	(6)	Shanghai
10.	(8)	Katarkis
11.	(11)	Tetris
12.	(-)	Holiday Maker

Wie erwartet ist es ELITE, dem TOP 12-TIP der Januarausgabe, gelungen, die Spitze der Hitparade zu erklimmen, und auch DUNGEON MASTER, der Tip des letzten Monats, schaffte immerhin den Einstieg auf Platz 4 und ist damit auch Aufsteiger des Monats. Weitere Neueinstiege sind DRAGONS LAIR und HOLIDAY MAKER, die sich hoffentlich auch in Zukunst gut plazieren werden.

Auch diesmal sollten Sie bei der TOP 12 teilnehmen, denn es gibt wieder 12mal den TOP 12-TIP zu gewinnen. Dies ist in der Regel ein brandneues Spiel, das bei der Redaktion besonderen Gefallengefunden hat. Nach der Flut guter Spiele im letzen Heft war diesmal die Auswahl merklich geringer. Nichtsdestotrotz haben wir als TIP das Action- und Strategiespiel JEANNE D'ARC auserwählt. Lassen Sie sich überraschen, was es das nächste Mal sein wird - und vergessen Sie nicht, Ihre Postkarte rechtzeitig einzuwerfen.

Mitmachen kann jeder, der die TOP 12-Postkarte (bitte keine andere!) ausfüllt und an uns zurückschickt. Der Rechtsweg ist dabei ausgeschlossen. Einsendeschluß ist der 30. Februar 1989. Karten, die nach diesem Termin eintreffen, werden im nächsten Monat berücksichtigt.

Wir gratulieren den Gewinnern dieses Monats, die jeweils einmal den TOP 12-TIP JEANNE D'ARC gewonnen haben: M. Dominok, Duisburg / Axel Maurer, Scheidt / Jens Bohrmann, Norden / Jürgen Reimold, Schwaigern Gerhard Hörth, March-Hugstetten / Michael Pietsch, Bayreuth / K. Hägele, Stuttgart / Karin Neber, Hamburg Carsten Wartmann, Berlin / Oliver Eichhorn, Obernburg / Graf Johann, München / Rainer Dörpfeld, Leer

Jeanne d'Arc

Jeanne d'Arc versetzt Sie zurück ins 15. Jahrhundert. Frankreich befindet sich seit über 100 Jahren im Krieg gegen die Engländer.
Am 25. Oktober 1415 wird die französische Armee vernichtend geschlagen. Daraufhin versuchen die mächtigsten Fürsten des Landes, ihre Macht auszubauen, das Resultat ist ein blutiger Bürgerkrieg; auch vor Bündnissen mit den Engländern wird nicht zurückgeschreckt. Frankreich, das mittlerweile von den Engländern regiert wird, holt zum letzten vernichtenden Schlag gegen das Königreich Frankreich aus und belagert Orléans, dessen Fall das Ende bedeuten würde. Doch eine 17jährige Bäuerin, Jeanne d'Arc, macht sich auf, den Thronfolger und Frankreich von den Engländern zu befreien. Eine kleine Armee wird ihr unterstellt, und so macht sie sich auf, Orléans zu befreien. So viel zur Hintergrundgeschichte, die den historischen Tatsachen entspricht. Dem Spieler ist es jetzt in die Hand gelegt worden, das weitere Schicksal Frankreichs zu bestimmen.

Die Karte von Frankreich gibt Aufschluß über die Lage der feindlichen Truppen.

Nach einer gewonnenen Schlacht wird ein sehenswertes Bild gezeigt.

Jeanne d'Arc ist ein Strategiespiel mit einigen Actioneinlagen, eine Mischung aus Defender of the Crown und Kaiser. Gespielt wird mit Maus, Tastatur und/oder Joystick. Das in deutscher Sprache erhältliche Programm wird mit deutscher Anleitung ausgeliefert, aus der der Spieler viele hilfreiche Informationen erhalten kann.

Zu Beginn des Spiels muß Orléans befreit werden, damit der König in Reims gekrönt werden kann. Dann erst können politische Entscheidungen getroffen werden, hierbei stehen Diplomatie, Handstreich, Justiz und einige Menüpunkte mehr zur Auswahl. Es können Bündnisse geschlossen, Intrigen vorbereitet, Spionage betrieben, Feldzüge vorbereitet und Steuern eingetrieben werden. Viele Charaktere stehen hierbei für die einzelnen Aufgaben zur Verfügung, nicht jeder ist als Feldherr geeignet oder als Diplomat. Ein kleiner Tip: Die am Anfang genannten Personen sollten notiert werden, Freund und Feind können dann wenigsten zu Beginn auseinandergehalten werden. Ohne schriftliche Notierungen kommt der Spieler ohnehin kaum aus, zu komplex ist das Spiel. Zu diesem Zweck liegt auch eine Landkarte von Frankreich bei, in der die verschiednen Stellungen festgehalten werden können.
Das Spiel besticht durch ausgezeichnete Grafik, oft werden digitalisierte Bilder als Hintergrund herangezogen. Der Sound hingegen ist etwas kümmerlich. Zu monoton ist das Kampfgeschrei. Jeanne d'Arc ist eher für ruhige Gemüter geschrieben. Einige Minuspunkte besitzt das Spiel aber auch. Der Spieler muß mit langen Ladezeiten rechnen. Obwohl die zwei Disketten keinen Kopierschutz besitzen, kann Jeanne d'Arc nicht so einfach auf eine Festplatte kopiert werden. Unsinnigerweise haben die Programmierer die direkte Adressierung der Laufwerke gewählt, anstatt die Disketten mit ihren Namen anzusprechen. Ein weiterer Minuspunkt ist die Tatsache, daß keinerlei Ereignisse rückgängig gemacht werden können. Ist ein Menüpunkt angewählt, gibt es kein Zurück mehr. Neben dieses Minuspunkten ist Jeanne d'Arc aber ein ausgezeichnetes Spiel, das viele interessante Spielstunden verspricht.

AK

Vertrieb: Rainbow Arts

FALCON

Haben Sie 35 Millionen Dollar?

... wenn nicht, sollten Sie wenigstens knapp 100 DM investieren, um ins Cockpit einer F16 Falcon zu gelangen. Möglich macht es die britische Softwarefirma Spectrum Holobyte/Mirrorsoft, denn aus ihrer Werkstatt stammt der neue Flugsimulator FALCON. Es handelt sich dabei um die wahrscheinlich ausgefeilteste Simulation, die für absehbare Zeit auf dem Markt erhältlich sein wird.

Allein das sehr aufwendig und detailliert gemachte Handbuch ist die Anschaffung wert. Es bietet nicht nur eine ausführliche und didaktisch geschickte Beschreibung des Programms, sondern darüber hinaus noch interessante Informationen für denjenigen, der sich für das Fliegen in überschallschnellen Militärjets begeistert.

Neben genauen Beschreibungen der F16 und ihres Simulationsgegners MiG 21 findet sich eine Darstellung sämtlicher von der F16 zu benutzenden Waffensysteme. Schon in der Einleitung wird einem der Geist, mit dem man diese Waffen einzusetzen hat, klargemacht. Leichtgemacht wird einem dies allerdings nicht, denn schon beim Ausrüsten der Maschine muß man sich über die Erfordernisse und zu erwartenden Gefahren der gewählten Mission Gedanken machen. Da die Flugeigenschaften sich mit der Zuladung rigoros ändern, können bei Überladung fatale bis letale Situationen entstehen. Glücklicherweise handelt es sich jedoch nur um eine Simulation, so daß man bei wenig erfolgreichem Ausgang Zeuge seiner eigenen Beerdigung werden und die Staffelkameraden bei der Ausführung einer "Missing Man" genannten Flugfigur - als Ehrung des ums Leben gekommenen Piloten - beobachten kann. Damit man nicht sogleich durch solch tragische Umstände die Lust am Spiel verliert, ist die Option des First Lieutenant eingebaut, bei welcher man weder abgeschossen werden kann noch abstürzt. Wenig zum Realitätscharakter der Simulation trägt in diesem Stadium die Möglichkeit, durch-Berge zu fliegen und auf Seen zu landen, bei.

Der Schwierigkeitsgrad und die Realitätsnähe können vom Spieler durch Auswahl eines von fünf militärischen Rängen - gestaffelt von First Lieutenant über Captain, Major, Lieutenant Colonel bis Colonel - bestimmt werden. Um ins Spiel zu gelangen, muß man als erstes über einen sogenannten Duty Roster - das ist ein Verzeichnis sämtlicher "Piloten", die jemals an der jeweiligen Simulation teilgenommen haben, wobei Details wie zum Beispiel ein Bild von den Lieben daheim an der Wand des Einsatzbesprechungsraums besonders erwähnenswert sind - gehen. Auf diesem Duty Roster - der sehr schön in Form eines Clipboard ausgeführt ist - kann man seinen Namen bzw. den Codenamen (z.B. Maverick, Iceman etc. - der Fantasie sind keine Grenzen gesetzt) eintragen. Bei dem abgespeicherten Namen werden die Meriten - abgeschossene Feindflugzeuge, gebombte Ziele, Orden etc. - und der jeweilige Stand des Ranges vermerkt.

Nach dem Duty Roster kann man die nächste zu spielende Mission und die Rangstufe, auf der diese laufen soll, wählen. Bei der "Milk Run" genannten Übungsmission kann man zusätzlich noch die Anzahl der zu erwartenden Feindflugzeuge einstellen. Danach gehts ans Zuladen der Bewaffnung. Auf der First Lieutenant-Stufe hat man ständig sämtliche Waffen zur Verfügung, ohne daß die Flugeigenschaften dadurch beeinflußt würden. Für alle anderen Ränge muß man die Bewaffnung unter Berücksichtigung des Gewichts der einzelnen Waffensysteme und der Anzahl der dafür vorgesehenen Befestigungen an Flugzeugrumpf und -flügeln auswählen. Dabei steht einem der Sergeant der Bodencrew mit Rat und Tat zur Seite. Endlich gelangen wir ins äußerst detaillierte Cockpit. Als erstes blicken wir durch die Frontscheibe, auf die das sogenannte HUD Display - eine visuelle Darstellung bestimmter Instrumente und Waffenelektronik vor den Augen des Piloten - projiziert wird. Darunter finden sich die Anzeigen für die Turbinenumdrehungen, der künstliche Horizont, ein Angriffs- und ein Defensivradar und mehrere andere technische Geräte. Ob diese alle aktiviert sind, hängt vom derzeitigen Rang des Piloten - und damit von der Realitätstreue der Simulation - ab. Mit Hilfe verschiedener Zahlentasten kann man einen Blick nach links, rechts und nach hinten - dazu müßte man allerdings schon ein echter Wendehals sein, da man hier einen Direktblick auf die Kopfstütze des Sitzes hat - riskieren. Auf der linken Seite findet sich der Gashebel - stilecht mit behandschuhter Hand - und einige andere Instrumente. Rechts findet man den kurzen Steuerknüppel - ebenfalls in behandschuhter Hand - und die Anzeigetafel für technische Defekte der Maschine. Die Turbine wirft man durch Betätigen einer Taste an. Danach geht sie hoch auf 60 Prozent Schub. Befindet man sich nicht in optimaler Startposition - das ist die nach Norden führende Startbahn - gibt man etwas mehr Schub, löst die Radbremsen und fährt auf dem Boden in die gewünschte Position. Lenken kann man entweder mit der Tastatur, der Maus oder wie echte Piloten mit dem Knüppel, je nachdem, was man vorher durch das Menü angewählt hat. Möchte man nun endgültig starten, geht man mit dem Schub auf mindestens 80 Prozent, löst wiederum die Radbremsen und saust auf der Rollbahn dahin. Bei einer Geschwindigkeit von mindestens 150 Knoten/h kann man abheben. Am besten fährt man schon jetzt das Fahrgestell ein, da man später durch das Blinkzeichen nur unnötig genervt wird. Auf das Angriffsradardisplay kann man eine Landkarte des Spielgebiets - diese ist in neun Quadranten eingeteilt, von denen zwei eigenes Territorium, die

restlichen Feindgebiet darstellen - projizieren. Nach einiger Übung kann man sich anhand markanter Punkte in der Landschaft orientieren. Die Landschaft ist grafisch leider auch nicht ansprechender gelöst, als man das von anderen Simulationen dieser Art kennt. Bodenziele kann man entweder mit den kameragelenkten Maverick - AGM 65B - Raketen, den 2000 lbMK-84-Bomben oder den speziell gegen Flugplatzrunways entwickelten Durandal Bomben angreifen. Gegen feindliche Flugzeuge oder SAM-Raketen - Surface to Air, von denen drei verschiedene Typen auftreten, denen jeweils verschieden begegnet werden muß - stehen einem außer der fest eingebauten M61-A1 Vulcan Kanone AIM-9J Sidewinder- und AIM-9L Sidewinderraketen zur Verfügung. Die komplizierte Handhabung dieser Bewaffnung wird zwar sehr genau erklärt, erfordert aber trotzdem erhebliche Übung. Nützlich gegen feindliche Radarerfassung erweist sich auch die ALQ-131 ECM-Zusatzladung. Allerdings sollte man hier gut überlegen, ob man dem feindlichen Radar nicht durch den so umstrittenen Tiefflug zu entgehen versucht und statt dessen drei externe Treibstofftanks von je 2304 lb bestem Kerosin mitführt. Sollte es zum Luftkampf kommen, kann man diese - wie übrigens alle anderen Außenladungen auch - zwecks besserer Manövrierfähigkeit abwerfen.

Insgesamt können siebzehn Missionen verschiedenen Schwierigkeitsgrades - für deren erfolgreichen Abschluß es jeweils unterschiedliche Belobigungen in Form von Ordensbändern gibt - absolviert werden. Jeder Mission werden Punkte von 1-15 zugeordnet, die mit einem Faktor des jeweiligem Ranges - Captain x 2 bis Colonel x 5 - multipliziert werden. Für jedes erfolgreich gebombte Ziel gibt es einen, für jeden abgeschossenen Feind und jede erfolgreiche Landung zwei weitere Punkte. Schafft ein Pilot alle Missionen erfolgreich und lebend - auf dem Duty Roster erscheint neben dem gespeicherten Namen der aktuelle Status [aktiv, verschollen (MIA - Missing in action), getötet (KIA - Killed in action) und unehrenhaft entlassen (busted)] des Piloten - erhält er weitere 3000 Punkte. Für jede schon erhaltene Medaille gibt es wiederum zusätzliche Punkte von 1-200. Die einzelnen Medaillen - Purple Heart, DFC (Distinguished Flying Cross), Silver Star, AFC (Air Force Cross) und die heißbegehrte Medal of Honour - erfordern bestimmte Qualifikationen. Bei nicht erfolgreichem Abschluß des Auftrags erscheint jeweils eine Sequenz von sogenannten "Schnappschüssen", die zwar sehr schön gemacht sind, aber auf die Dauer etwas langweilen. Die Schnappschüsse stellen die Reihenfolge eines Unglücks - z.B. Raketentreffer, Ausstieg mit dem Schleudersitz, Hängen am Fallschirm und schließlich die erlösende Rettung durch den Hubschrauber - dar. Die Reihenfolge kann sich ändern, so daß statt des Aussteigens mit dem Schleudersitz ein unkontrollierbares Trudeln mit folgendem Totalverlust des glücklosen Piloten resultiert. Der letzte, sehr umfangreiche Abschnitt des Handbuchs beschäftigt sich mit Luftkampfmanövern für Fortgeschrittene. Sehr interessant sind hier die genauen Erklärungen zu den bei Kurven- und Sturzflügen auftretenden Gravitationskräften. Die F16 ist das erste Jagdflugzeug der Welt, das so enge Kurven fliegen kann, daß Kräfte von +9g auf Mann und Maschine wirken. Man sollte sich diesen Abschnitt aufmerksam durchlesen, da es auch in dieser Simulation bei zu großen positiven Gravitationskräften zur Visusverminderung bis hin zum Blackout - der Bildschirm flackert und wird schwarz - kommen kann. Bei zu großen negativen Gravitationskräften kommt es zum sogenannten Redout durch Platzen von Blutgefäßen in den Augen - der Bildschirm erhält eine rötliche Färbung. Bei einem Megabyte Speicherplatz kann man bei Flugmanövern eine Black Box mitlaufen lassen, mit deren Hilfe man sich später alle eigenen und gegnerischen Manöver noch einmal ansehen kann. Richtig spaßig wird die Angelegenheit bei Kopplung zweier Computer, die in der Anleitung für verschiedene Geräte erklärt wird. Bonbons des Handbuches finden sich am Schluß in Form einer weiterführenden Bücherliste zum Thema Luftkampf und eines Lexikons von Pilotenausdrücken.

Fazit: Im Vergleich zur besten bisherigen Militärjetsimulation F/A 18 Interceptor ist FALCON, was die Genauigkeit und Realitätstreue betrifft, um mehrere Klassen besser. Der Vergleich läßt sich mit der F16-Option von Interceptor umso besser treffen. Ein gravierender Nachteil von FALCON ist das Setting in Air Force-Umgebung. Abgesehen vom Fehlen jener vornehmen Atmosphäre der Marinefliegerei - die man nicht zuletzt aus dem Hollywoodprodukt TOP GUN zu kennen glaubt - enttäuscht das Fehlen jenes exquisiten Nervenkitzels, den selbst eine simulierte Trägerlandung darstellt. Darüber kann leider die auch recht komplizierte Landetechnik von FALCON nicht trösten. Dessweiteren vermißt man nach einiger Zeit das simulierte Aussteigen mit dem Schleudersitz und das folgende Fallschirmgleiten, wenn auch das Interceptor-Strichmännchen hoffentlich noch nicht die endgültige Lösung dieses Themas darstellt. Der Hinweis auf Anregungen von Seiten der Benutzer zu Anfang des Handbuchs von FALCON zur Verbesserung des Programms läßt da vielleicht hoffen. Ansonsten hält man mit FALCON unserer Meinung nach die beste Simulation - nicht nur für Flugzeuge, sondern überhaupt - die es bisher gibt, in Händen und wird für die knapp 100 investierten DM lange Zeit eine interessante Aufgabe haben.

Wer sich an dem kriegerischen Szenario stört, wird sich ärgern, bis hierher überhaupt gelesen zu haben. Das Für und Wider dieses Themas wurde an anderer Stelle schon mehrfach erörtert, so daß wir hier darauf verzichten zu können glauben.

Christian Günther Oliver Siebenhaar

Anbieter: CDC, Bad Homburg
Preis: 99.-

Hot Ball

'Fußball ist vorbei! Hotball schlägt die Grenzen des meistgespielten Spieles der Welt weit zurück....'(Zitat Cover). Tatsächlich schlägt Hotball alles, was uns bisher auf dem Amiga an Fußballsimulationen geboten wurde. Ich würde sogar noch weiter gehen - Hotball überschreitet die Grenzen der Zumutbarkeit.

Beginnen wir mit der Anleitung. Laut Hersteller ist sie in Englisch. Unsere Nachforschungen sind mittlerweile soweit gediehen, daß wir den Weg der Übersetzung aus dem Französischen ins Japanische und danach ins Finnische rekonstruieren können. Wie die Anleitung dann allerdings mit der chinesischen Sprache kollidierte, aus der sie direkt ins Hebräische und von dort ins Englische übersetzt zu sein scheint, haben wir bisher noch nicht herausgefunden.

Das Team: : "...Marc Bousquet: He spent over three years at the option page...". Kann ich gut verstehen. Ich habe vergleichbar lange gebraucht, um damit zurecht zu kommen.

"...Michel Desangles: He stopped the psycho-freak adaption of Beethovens ninth he was planning to do the music. But he spends more time on stadiums than on keyboards, because his real hobby is football..." Welch ein Trost zu sehen, daß Michel am Ende erkannt hat, wofür er sich besser eignet, hoffentlich.

"...Marc Djon: He was often compared to Michelangelo and Andy Warhol, for the subtleness of his drawings, precision of his graphics and for the smooth shades of his colours. His passion: computing..." Dieser Junge hat wirklich Talent. Schade, daß gerade er sich als Einziger nicht für Fußball interessiert.

"...Fracois Gullerme: He is a great Optimizer. He can work two months to gain two milliseconds in a routine..." Leider scheint dieser tolle Programmierer nur insgesamt sieben Wochen dabei gewesen zu sein.

"...Rene Bouraqul: This one is not on the picture, because he stumbled over a passing ball and had a middle finger damage, right after he finshed to design the hall of fame..." - zu spät! "...Antoine Matthieu: He did almost everything: Animation, players routines, scrollings, nothing is to hard for him..." - nur das Ergebnis, Rene, das war vielleicht ein wenig zu hart für uns.

"...Phillipe Chastel: He managed to take small bits of program from everyone and mix them together..." -dies erklärt vieles- "...the only place he feels secure is on a stadium..." - falls die Fans keinen Amiga besitzen.

"Xavier Cador: He ist the tall guy and therefore the boss ... A sort of miracle. His only passion football..." Eigentlich war ich der Meinung, daß in Frankreich die gleiche Art Fußball gespielt wird wie in der Bundesrepublik und in anderen Ländern. Mittlerweile sind mir ernste Zweifel gekommen.

Das Spiel ruckt und zuckt (Fachausdruck: scrolling), die Figuren torkeln über den Platz als wären sie volltrunken und laut Ballabfrage liegt das Tor knapp neben der Eckfahne.

CALIFORNIA GAMES

Lange angekündigt, auf anderen Rechnern schon seit geraumer Zeit veröffentlicht, ist CALIFORNIA GAMES aus dem Hause EPYX jetzt endlich auch für den AMIGA erhältlich. Sechs Diziplinen oder Freizeitsportarten gilt es diesmal zu bewältigen:

SURFING, FOOT BAG, SKATE BOARDING, ROLLER SKATING, FLYING DISK und BMX. Wie von EPYX gewohnt, erscheint zunächst ein Menü, in dem verschiedene Einstellungen vorgenommen werden können. Disziplinen können trainiert oder im globalen Wettkampf ausgesucht werden. Auch kann sich der Spieler die besten Ergebnisse anschauen. Bis zu acht Spieler können gegeneinander antreten und um Punkte bzw. Pokale kämpfen. Zuvor muß sich jeder Spieler noch einen Sponsor zulegen, dann geht es los.

Alle sechs Disziplinen werden mit dem Joystick gesteuert, wobei einige recht schwer zu handhaben sind. Es gehört schon ein wenig Übung dazu, eine Disziplin zu beherrschen. Die Grafik ist als durchschnittlich zu betrachten, könnte aber für AMIGA-Verhältnisse besser sein. Eine bessere Konvertierung für den AMIGA wäre zu wünschen. Die Animation der Sprites ist hingegen gut. Der Sound ist bis auf ein paar gute Einlagen recht mager.

Sportspiele besitzen ihren Reiz, besonders wenn sie gleich mehrere Wettkämpfe beinhalten und mehrere Personen gegeneinander antreten können. Ein echtes 'Wettkampffeeling' kann durchaus aufkommen. AMIGA-Anwender, die Programme wie Summer-, Winter- oder World-Games mögen, werden auch Gefallen an California Games finden.

Bei California Games stehen sechs Disziplinen zur Auswahl.

CDC
Louisenstr. 115
6380 Bad Homburg
Tel. 06172-20799

RAC Lombard-Rallye

Auf den Spuren Walter Röhrls

Schon Nicki Lauda erkannte den Blödsinn des ständigen "im Kreis herum"-Fahrens. Die Alternative zu Rundstreckenrennen ist, auch auf dem AMIGA, das Rallyefahren.

Alternativen zum monotonen Rundkurs hat die "Lombard-Rallye" genügend aufzuweisen. Fünf Etappen mit je drei Wertungsprüfungen gilt es mit einem Sierra RS des Ford-Edelrennstalls Cosworth zu bewältigen. Wie die echte Rallye führt auch die Computersimulation quer durch England. Küstenstraßen, Bergprüfungen und Waldstrecken müssen teilweise bei Nacht und Nebel durchgefahren werden. An der eigentlichen Rallye über sämtliche Wertungsprüfungen kann nur teilnehmen, wer sich zuvor in sämtlichen Teiletappen bewährt hat und zumindest einmal unter dem Beifall der Zuschauer auf dem Treppchen stand.

Doch nun zum Start:

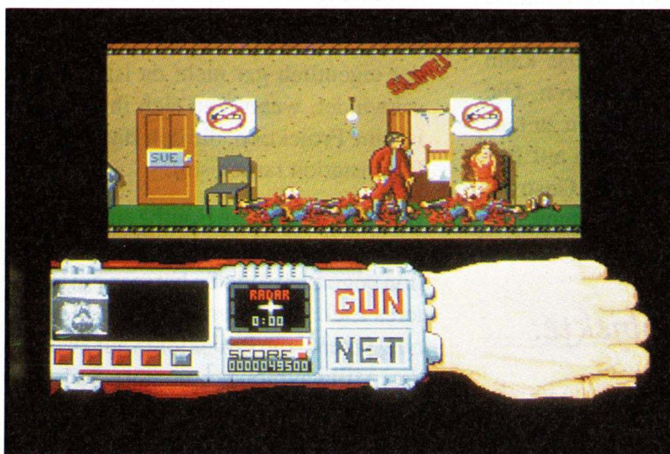
Bevor sich die Starterflagge hebt, hat der Pilot noch die Möglichkeit, einen ungestörten Blick auf das Gebetbuch des in good old England links sitzenden Beifahrers zu werfen. Dann wird es ernst. Der noch serienmäßige Cosworth reagiert spontan auf den Druck des Joysticks nach vorne, und schnell sind die vier Gänge über den Feuerknopf durchgeschaltet. In den Kurven empfiehlt es sich, zunächst das Gas wegzunehmen oder gar zurückzuschalten, damit der Sierra anschließend wieder schnell auf Touren kommt. Drei von acht Kontrollinstrumenten zeigen dem Fahrer den Verschleiß von Fahrwerk, Motor und Reifen an. Wichtiger ist jedoch die Beobachtung des Drehzahlmessers, denn unter viertausend Touren tut sich in den beiden hohen Gängen nicht viel. Der wichtigste Blick führt allerdings geradeaus, viel Zeit für eine Beobachtung des Cockpits bleibt nicht. Vorne ist schon der zuvor gestartete Peugeot 205 zu sehen. Aber Vorsicht: Ausflüge in das Gelände beschädigen das Fahrwerk; bei Unfällen fliegen dem Fahrer die Zusatzscheinwerfer um die Ohren - und das schlägt sich nicht nur in der Gesamtzeit nieder, sondern auch auf die Werkstattrechnung. Nebem dem Fahren erfordert die Lombard-Rallye aber noch andere Eigenschaften vom Fahrer:

Als sein eigener Teammanager muß er sich auch um die Finanzierung kümmern. Geld gibt er als Newcomer in erster Linie für die bei Geländeausritten entstandenen Blessuren aus. Mit zunehmender Fahrkunst wird die regelmäßige Wartung des Wagens billiger, dafür wird dann im Tuningshop die eine oder andere Mark landen. Neben umfangreichen Extras können Allradantrieb und stärkere Motoren eingekauft werden. Einnahmen lassen sich in Interviews - einem Trivial Pursuit des Rallyesports - durch richtiges Beantworten der gestellten Fragen erzielen und natürlich durch Etappensiege. Lombard-Rallye hinterläßt einen guten Gesamteindruck. Fahrsimulation und Sound sind gut konzipiert, ohne jedoch neue Perspektiven für den AMIGA bieten zu können. Durch das Einbringen der Finanzierung wird die Lombard-Rallye zu einer Simulation, die in keiner guten Sammlung fehlen sollte. Einziger Kritikpunkt ist die teilweise mangelhafte deutsche Übersetzung.

Michael Döring/cpl

Techno Cop

muß sowas sein?



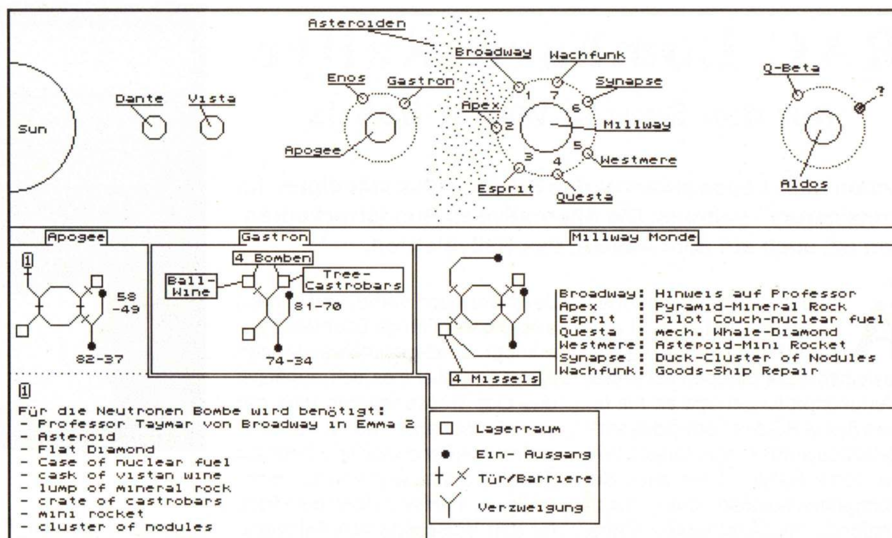
Ich möchte im Voraus sagen, daß ich wirklich nichts gegen GUT GEMACHTE Baller- oder Actionspiele habe. Aber was zuviel ist, ist zuviel. In Techno Cop übernimmt der Spieler die Rolle eines Polizisten in der - hoffentlich nicht allzu nahen - Zukunft.

Die Aufgabe des Spielers ist es, "mit dem Abschaum aufzuräumen, die Straßenbanden zu zerschlagen und die Aussteiger zu eliminieren, die die Stadt verschmutzen". Dies gestaltet sich auf dem Bildschirm folgendermaßen: Man fährt mit seinem mit Waffen bestückten Auto zum Tatort, "jagt alles in die Luft", was sich bewegt (ruckelnde Autos, Bäume und Motorräder), und kommt dann irgendwann auch an. Nun beginnt der zweite Teil dieses sogenannten Spiels. Man befindet sich in einem Gebäudekomplex und muß versuchen, über Fahrstühle zu dem vom Computer angegebenen "Oberschläger" zu kommen und - gemäß Auftrag - "sofort das Feuer eröffnen, sobald man diesen zu Gesicht bekommt". Die verdächtigen Häuser sind natürlich mit allerlei Gegnern bevölkert, die es zu "vernichten" gilt. Hat man diese oft genug getroffen, verwandeln sie sich in unförmige Blutklumpen, zucken noch ein paarmal und bleiben dann liegen. Diese Szenen sind so detailliert und ekelerregend dargestellt, daß dieses Spiel wohl (mit Recht und meiner Zustimmung) auf dem Index landen wird. Hat man den Übeltäter unschädlich gemacht, geht das Spiel immer im gleichen Stil weiter. Die Grafik von Techno Cop ist mehr schlecht als recht, einzig die Animation des Einzelkämpfers und seiner Gegner ist wirklich gut. Der Sound beschränkt sich auf Motorgeräusche, Schüsse und Todesschreie (was aber der Zielgruppe dieses Spiels genügen sollte). Abschließend ist zu sagen, daß allen Mitgliedern von Wehrsportgruppen dieses Spiel uneingeschränkt empfohlen werden kann. Ach ja, alle Sätze oder Worte dieses Tests, die in Anführungsstrichen gesetzt sind, sind Zitate aus der Anleitung.

Sven Stillich

STARGLIDER II

Tips



Auf Apogee empfiehlt es sich, zuerst einmal einen Eingang ins unterplanetarische Höhlensystem zu finden. Dort sollte man sich in das im Plan mit 1 bezeichnete Lager begeben, wo man dann von der Untergrundorganisation eine Liste von 9 Dingen, die es aufzutreiben gilt, bekommt (Dieser Schritt ist eigentlich nur notwendig, wenn man die chronologische Reihenfolge wahren will. Ansonsten kann man gleich bei Spielbeginn mit dem Einsammeln der Gegenstände beginnen.)

Es gibt insgesamt drei Möglichkeiten, die Gegenstände zu bekommen.

Erstens kann man von Planet zu Planet fliegen, dort umherfliegen, bis man den gesuchten Gegenstand findet, und ihn einsammeln.

'Cluster of nodules' : findet man auf Dante, meistens in der Nähe von Vulkanen.

'Cask of vistan wine' : findet man logischerweise auf Vista.

'Flat diamond' : befindet sich ebenfalls auf Vista, in der Nähe der 'blow holes'. Nicht überrascht sein, sie hüpfen auf und ab.

'Asteroid' : ist nur im Weltraum zu finden, vor allem zwischen Apogee und Millway.

'Professor Taymar' : fährt auf Broadway in einem Gefährt namens *Emma 2* umher. Es gibt dort zwar viele Emmas, aber egal, welche man einsammelt, es ist immer die von Professor Taymar. Man sollte aber zuerst im Tunnelsystem die Kammer aufsuchen, in der man einen Hinweis auf den Aufenthaltsort des Professors bekommt. Mit etwas Glück wird ein Geschenk erwähnt. Sammelt man daraufhin den Professor ein, bekommt man unter Umständen einen *'Energy-Cube-Launcher'*. Dieser ist zwar nicht unbedingt notwendig, aber man muß jetzt weniger oft treffen, um denselben Effekt zu erzielen.

'Lump of mineral rock' : ist auf Q-Beta zu finden.

'Case of nuclear fuel' : liegt auf Aldos herum.

'Mini rocket' : kann man auf fast allen Planeten und Monden finden.

Es fällt wohl auf, daß es auf keinem Planeten oder Mond ein *'crate of castrobars'* gibt. Diese findet man eigentlich nur im Laderaum von Piratenschiffen. Andere Dinge wie *'vistan wine'*, *'mineral rock'* oder *'cluster of nodules'* haben sie auch manchmal geladen. Wird man also in einen Kampf mit den Piraten verwickelt, ist es ratsam, genau zu zielen und immer nur einmal zu schießen. Was dann vom Piratenschiff übrigbleibt, sollte man einsammeln und, sofern man es brauchen kann, behalten. Das kann unter Umständen viel Zeit sparen. Die dritte Möglichkeit besteht darin zu tauschen. Getauscht werden kann auf fast allen Millwaymonden und auf Gastron. Wo man was gegen was eintauschen kann, ist auf der Karte eingezeichnet.

Die Tauschobjekte:

'Pyramid' - findet man auf allen möglichen Planeten u. Monden.

'Pilot Couch' - habe ich bis jetzt noch nicht gefunden. Unter Umständen bekommt man eine, wenn man auf einem Planeten ein Raumschiff abschießt und die Rettungskapsel einfängt. Das habe ich allerdings noch nicht geschafft.

'Whale' - begegnet einem manchmal im Weltraum, vor allem dicht über Millway.

'Asteroid' - siehe oben.

'Duck' - findet sich meistens in der Nähe eines 'Whale'.

'Tree' - Enos ist ein Waldmond. Also findet man dort haufenweise Bäume.

'Bouncing Ball' - hüpft auf Vista umher.

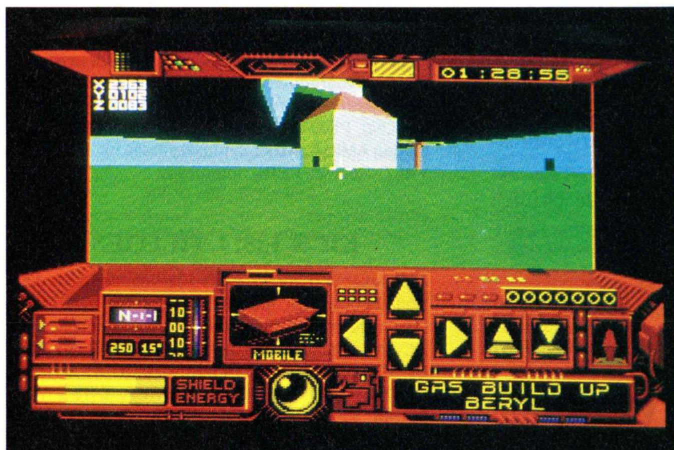
Der gelbe Mond, der um Aldos kreist, der eigentlich gar nicht da ist und verschwindet, wenn man sich ihm nähert, ist eine Projektion, die den Bauplatz der Raumstation tarnen soll.

Hat man endlich die Bombe an Bord (!!), sollte bereits mit dem Bau der Raumstation begonnen sein. So hat man ein leichteres Ziel. Dann werden mit den *'bouncing bombs'* die Projektoren auf zwei bis drei Millwaymonden zerstört. So ist die Raumstation nicht mehr geschützt und kann mit der Bombe zerstört werden.

Nachdem man das geschafft hat, geht das Ganze wieder von vorne los.

ANDREAS PAUL

Driller



Ein neues Vektorgraphikspiel für den AMIGA, bei dem entsprechend dem jetzigen Qualitätsstandard die Flächen natürlich alle ausgefüllt sind. Doch was bringt das Spiel sonst so? Sie haben mal wieder einen Planeten vor dem Untergang zu retten. Auf einem der Monde des Planeten wurde sehr viel Bergbau betrieben. Das Volk, das hier gearbeitet hat, ist aber von diesem Mond geflohen. Mit der Zeit haben sich große Gasmengen in den zurückgelassenen Gängen und Höhlen gesammelt. Dadurch besteht die Gefahr der Explosion des Mondes, was zur Folge hätte, daß Tausende von Bewohnern des Planeten Evath sterben müßten und der Planet in eine andere Bahn geworfen würde. Da Wissenschafter einen Meteor entdeckt haben, der auf den mit Gas vollgepumpten Mond zufliegt, und es sich nur noch um Stunden handeln kann, bis der Meteor einschlägt, müssen die Gase möglichst schnell aus dem Mond entfernt werden.

Dies ist, wie könnte es anders sein, Ihre Aufgabe. Sie werden auf den Planeten gebracht und haben anfangs nur eine Grabungssonde, in der Sie sich befinden, zur Verfügung. Mit ihr fahren Sie dann auf der Mondoberfläche herum und machen an verschiedenen Stellen Probebohrungen. Konnten bei einer Bohrung mehr als 50% Gas abgepumpt werden, ist der Sektor, auf dem Sie sich befinden, sicher. Sind alle der achtzehn Sektoren gesichert, haben Sie Ihre Mission erfüllt. Um es Ihnen nicht allzuleicht zu machen, gibt es an einigen Stellen Selbstschußanlagen, zerstörend wirkende Energiefelder, Tretminen u.s.w.. Außerdem erscheinen manchmal Ufos, die Sie mit gezielten Laserstrahlen ganz schön in Schwitzen bringen. Bei der Steuerung haben Sie die Möglichkeit zwischen Tastatur und Maus zu wählen. Benutzt man beide Steuerungen gleichzeitig, kann man sich am besten bewegen und blitzschnell reagieren.

Fazit

Die Spielidee, einen fremden Planeten zu erforschen, ist zwar nicht so neu, jedoch gut umgesetzt. Auf wüste Raumschiffschlachten wurde verzichtet und mehr Wert auf taktisches Überlegen gelegt. Driller ist sicherlich im oberen Bereich der Vektorgraphikspiele einzuordnen, obwohl die 3D-Animation nicht ganz so ruckfrei ist (wie zum Beispiel bei Interceptor).

(Florian Du Bois)

Tips

- Es lohnt sich nach einem versteckten Flugkörper zu suchen, mit dem man den Planeten schneller untersuchen kann.
- Befinden sich irgendwo Kabel, sollte man versuchen, sie kaputtzuschießen, da sie meistens Schußanlagen mit Energie versorgen.
- Kommt man in einen Raum, in dem sich an der einen Wand drei Zeichen befinden, sollten sie so verändert werden, daß sie verschieden sind.
- Wichtig ist es auch, auf unscheinbare Würfel oder Wände zu schießen, weil sich dadurch geheime Türen öffnen oder Wände verschwinden können.

INSERENTEN-VERZEICHNIS

KICKSTART PU

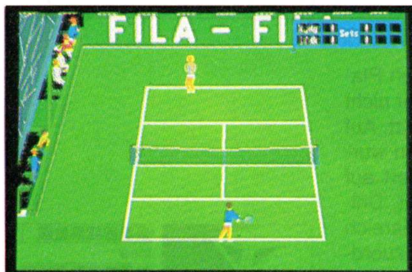
Hallo PD-Fans,

besonders empfehlen möchte ich Ihnen in diesem Monat die Source-Codes von Matt Dillion, die eine Fundgrube für jeden Programmierer darstellen und einen guten Einblick in die Programmierung des AMIGA geben. Weitere Höhepunkte sind AMIGASPELL -das erweiterbare Korrekturprogramm, NAG -der unverzichtbare Terminkalender und die Programme der UTILITY-Diskette. Sollten Sie Anregungen zu dieser Sammlung haben, dann würde ich mich über eine Mitteilung sehr freuen.g

Bis zum nächsten Mal
Markus Nerding

DIE NEUHEITEN

KICK 131: SPIELE

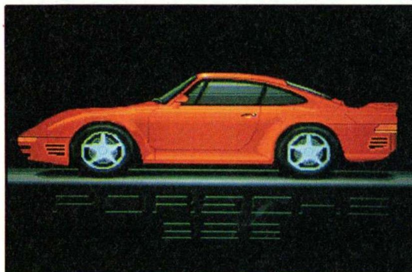


TENNIS: Tennisspiel für zwei Spieler mit Trainingsmodus, Grund-linien- und Netzspiel. Die Schlagstärke und Richtung können mit dem Joystick bestimmt werden. Erfordert 1MB Speicher. *Autor: Christian Dreke.*

KICK 132: SOUNDS

TGB SOUND DISK NO.1: Fünf Musikstücke (erstellt mit Soundtracker) werden vor dem Hintergrund von HAM-Bildern abgespielt. *Komponist: Thomas-Georg Brüggemann.*

KICK 133: GRAFIK-SHOW

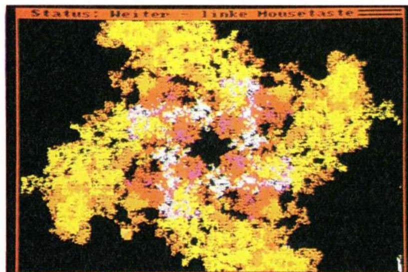


CRAZY-MOTOR-SHOW: gut gezeichnete Sportwagen und eigene Konstruktionen mit Angabe der technische Daten und interessanten Hintergrundinformationen. Mit digitalisierter Musik. 1MB Speicher erforderlich. *Autofan: I.K.*

KICK 134: BILDER

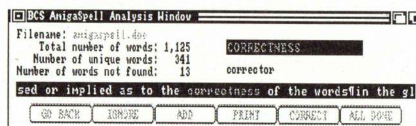
Viele gute Grafiken (SexyRobot, Roland SH-101, Sailboat, Max, Ferrari GTO, Dali, ST), Cartoons (Batman, Garfield) und digitalisierte Bilder (Supercollage von Marilyn, Mona Lisa, Alien).

KICK 135: GRAFIK



MANDEL 2.1: Mandelbrotgenerator für repititive Muster. Source in AmigaBASIC. *Autor: Michael M.*

KICK 136: ANWENDUNGEN



AMIGASPELL: Wörterbuch (Wortschatz: 9085 englische Wörter) zum Korrigieren von (ASCII)Texten. Falsche bzw. nicht vorhandene Wörter können ausgedruckt, im Text korrigiert (der Satzzusammenhang wird angezeigt!) oder übernommen werden. Schnell (Assembler) und speichereffektiv. Es kann leicht (wenn auch mit Arbeit) z.B. ein deutsches oder hessisches Wörterbuch erstellt werden.

NAG: der absolute, blinkende, schreiende, singende Terminkalender, damit Sie nie wieder einen Geburtstag, eine Fete oder den täglichen Feierabend vergessen. Sehr komfortable Eingabe, viele Möglichkeiten, immer griffbereit.

HP11: Emulation eines HP11C-Calculators mit allen Finessen.

KICK 137: SPRACHEN

Für alle Programmiersprachenbegeisterten ist hier eine kleine Fundgrube. **SCHEME**: ein Dialekt ähnlich Lisp, **PILOT**: eine Umsetzung dieser Sprache für den AMIGA, **ADA:** Syntaxkontrolle für ADA, mit YACC-Source, **PERL**: Interpreter

KICK 138: UTILITIES

Auf dieser und der KICKPD 139 befinden sich die Sourcecodes der meisten Programme von Matt Dillion, der sicherlich jedem PD-Anwender bekannt ist. Im einzelnen sind dies: **CONFIG V1.00, CLOCK V1.00, DME V1.31, DMOUSE V1.10, BACKUP V2.01, SUPLIB, LIBREF, DRES V1, DASM V2.11, FTOHEX, FILES V1.2, SHELL V2.10, FINDIT V1.00, LIBS V1.00, SCAT, ADDCR, REMCR, CMP**.

KICK 139: UTILITIES

Hier sind die restlichen Sources der Programmsammlung von Matt Dillion (siehe KICK 138).

KICK 140: UTILITIES

RAMCOPY: Kopierprogramm für ein Laufwerk und mind. 1MB Speicher
DISKSALV: rettet gelöschte Dateien
SPUDCLOCK: Uhrzeitansage
MACHII: Maustool, Hotkeys, Popcli usw.
MULTICALC: RPN-Taschenrechner
MRPRINT: Druckprogramm für Dateien, mit vielen Optionen
AFTTERM: Terminalprogramm
MFIX: patcht das Kopierprogramm MARAUDER und steigert damit die Geschwindigkeit erheblich

Machen Sie mit!

Möchten auch Sie selbstgeschriebene Programme der Allgemeinheit zur Verfügung stellen, so schicken Sie sie uns einfach zu (bitte mit Dokumentation auf Diskette!).

Als kleine Anerkennung können Sie sich dafür 5 Disketten aus unserem PD-Service auswählen.

MAXON-Computer GmbH
KICKSTART-Redaktion
PD-Einsendung
Industriestraße 26
6236 Eschborn

BLIC DOMAIN

KICK 130: ANWENDUNGEN

SUPERLIGA: Fußballergebnisverwaltung mit 577 Mannschaften aus 23 Ländern und 34 Ligen. Über 10000 Ergebnisse können gespeichert werden. Das Eingeben gestaltet sich sehr einfach. Aus den bisherigen Ergebnissen kann eine Prognose (Regressinsanalyse) für kommende Spiel und die Meisterschaft erstellt werden. Autor: Rolf Morlock

KICK 129: SOUNDS

Neue Kompositionen von Andreas Starr aus Gronau, erstellt mit Sonix. Effektvolle Stereostücke mit Player.

KICK 128: SOUNDS

Scores, Drumkits und Soundfiles für Dynamic Drums von New Wave Software.

KICK 127: GRAFIK

Mandelbrot-Show II: dies ist der zweite Teil der Reise durch die Welt der Mandelbrotgrafiken. Autor: R. Beck

KICK 126: GRAFIK

Mandelbrot-Show I: eine sehr informative Reise durch die Welt der Mandelbrotgrafiken. Jedes Bild wird ausführlich erläutert und das jeweils darauffolgende ist eine Vergrößerung eines Ausschnitts davon. Mit Mandelbrotgenerator MSE. Autor: R. Beck

KICK 125: GRAFIK

SLIDESHOW: Komfortables Diashow-Programm für DPaint-Bilder (320x200). Alle Bilder des aktuellen Direktories werden nacheinander auf Mausklick angezeigt. **DARK**: Copper-Demo, die viele Elemente eines Ballerspiels enthält. C-Source. **GLOBE**: Eine kleine drehende Weltkugel. PopUp-Menu, Source in C. **ICONS**: verschiedenste, animierte Icons (Disk, Drawer, Tool, Project, Garbage). Sehr originell und lustig!

KICK 124: ANWENDUNG

STARCHART: umfangreiches und interessantes Sternenprogramm mit unendlich vielen Einstellungen. Mit der Maus können Informationen über die Sterne abgerufen werden. **AIRFOIL**: ein einfacher Windkanalsimulator, verschiedene Einstellungen.

KICK 123: UTILITIES

Neue Utilities, die aus den aktuellen Einsendungen zusammengestellt werden. Hier gibt es immer ein riesiges Angebot, denn der AMIGA läßt viele Möglichkeiten für Hilfsprogramme.

KICK 122: MODULA (AMOK 12)

Auch auf dieser AMOK-Diskette gibt es wieder jede Menge Routinen und Programme: Modul zum Ansteuern des Fischertechnik-Interfaces, ein schnelles und sicherers HD-Backup, Disassembler für M2Amiga-Object-Files, ArraySort, Janus, Timer, StripText, ConvLibSym, MakeInLine, Diff, Tips zum M2-Compiler

KICK 121: MODULA (AMOK 11)

R.o.M.: ein komfortables Mathematikprogramm und Funktionsplotter.**TrackDiskSupport**: erleichtert Programmierung der TrackDisk. **M2Druka**: Druckutility für ASCII-Dateien. **Beep**: erzeugt akustisches Warnsignal.

KICK 120: UTILITIES

PCopy, HexCalc, Dux5 (CLI-Utility), SDBackup, FlamKey (Computerschloß), VirusX, SCT (Farbeinstellung von WB), Surveyor, HPMan (totale Kontrolle über den Laser)

KICK 119: AMOK #10

FlightDemo u. 3D-Demo (Echtzeitdemonstrationen bewegter räumlicher Grafik - super), FileRequester, SuperLists1.3, Speech (dt. Sprachausgabe), Feigenbaum.

KICK 118: AMOK #9

GraphicSupport (Makros für Graphics-Primitives/Intuition), PrintIt2.0 (Hardcopyroutine für Epson), M2Test (Dokumentation über Fehler bei ARRAYs mit Index BOOLEAN), TextFont (Laden v. Zeichensätzen), Sounds und Graphics.

KICK 117: AMOK #8

MemSystem (Speicherverwaltung), PrinterSupport (unterstützt Printer.device), IDCMP (Hilfsroutinen), Blitter (direkte Blitterprogrammierung), SpaceDemo (3-D Demo), IFF8SVXLoad (Laden/Abspielen gesampelter Sounds).

KICK 116: AMOK #7

Reminder (Wecker für Programmierer), WarpText (schnelle Textausgaberoutine), MuchMore (Softsroll), M2Test (Hinweise auf Fehler in den Bibliotheksmodulen), SoundTask, AVL-Bäume, Listen, Queues, Stack.

KICK 115: SPIELE

CHEES-BOXES ist das allseits bekannte "Käsekästchen"-Spiel, das man jedoch nicht mit dem folgenden verwechseln sollte. 2-4 Spieler, mausgesteuert, AMIGABasic. Autor: J.D. Mallander. **VIER GEWINNT**: zwei Spieler, mausgesteuert, AMIGABasic. Autor: Björn Kessels. **COSMIC**: Strategiespiel. Source in C. Autor: Carl Edman. **SOLITAIRE** löst beliebige Stellungen des bekannten Spiels. Source in C. Autor: Thomas Lagally **GRID IT**: spannendes Rennen für zwei Spieler. Benötigt Joystick, Source in C. Autor: Jörg Tuttas **LABYRINTH II**: Textadventure in Stil der INFOCOM-Spiele. Englisch, C.

KICK 114: BILDER-SHOW

Eine Grafik-Show mit Mandelbrotbildern und Colorcycling. Dadurch wird die Wirkung der eindrucksvollen Bilder hervorgehoben. Autor: Hermann Dörries, Wildeshausen

KICK 113: SOUNDS

Auf dieser Diskette ist der zweite Teil der KICKSTART SOUND DEMO III mit den Stereo-MIXES: Return to Defender, Dark Castle Mix, Chicago-Tune, Disco-Mix und Crazy Tune. Wichtig: Die Diskette läuft nur mit KICK 112!

KICK 112: SOUNDS

Zum dritten Mal haben unsere Soundspezialisten TOB & TWP zugeschlagen. Das Ergebnis ist die KICKSTART SOUND DEMO III mit einer Menge hervorragender Sounds auf zwei Disketten (siehe KICK 113). Mit von der Partie sind in der Rubrik CHARTS: Relax, Jean Michelle Jarre, Every Breath You Take, Axel F., Push, Tell It To My Heart und Blue Monday. Alles HiFi-Stereo mit einer extralangen Spielzeit.

KICK 111: BILDER

20 wunderschöne IFF-Grafiken von Frauenprofilen. Mit Diashow, sehenswert!

KICK 110: UTILITIES

MEMORYCLOCK: zeigt Uhrzeit, Datum, RAM- und Chip-Memory. **DATAMAKER, IFF-CONVERTER**: wandelt Dump-Format-Bilder in IFF-Format. Source zu allen Programmen in Assembler, Autor: Roger Fischlin.
SDBACKUP: Festplatten-Backupprogramm, das die komplette Platte in komprimierter Form auf Disketten abspeichert. Viele Optionen. **RUNBACK:** läßt Programme im Hintergrund laufen, damit sie keine CLI-Fenster benötigen.

KICK 109: SCHULE

VOK-BOY V1.0: Vokabeltrainer in AmigaBasic. Autor: Michael Hennemann
WORD: Fremdsprachenlernprogramm, das Begriffe und ganze Sätze abfragt. Mit umfangreichen Lektionen in Englisch und einigen in Französisch. Kann leicht erweitert werden. Verwaltet beliebige Sprachen, sehr komfortabel. Autor: Daniel Neukomm, Bern (Schweiz)

KICK 108: BILDER

Viele schöne Bilder zum Anschauen und Weiterbearbeiten.

KICK 107: ANIMATIONEN

Weitere Animationen von Tobias Richter: JET (Düsenjet-Landeanflug), RELIANT (ein Raumschiff im frontalen Anflug), NELSON (rotierendes Raumschiff). Die Animationen demonstrieren, wie eindrucksvoll die mit VideoScape erstellten Sequenzen sein können. Sehenswert!

KICK 106: ANIMATIONEN

Tobias Richter hat uns wieder einige seiner mit VideoScape 3D-Animationen geschickt: LOTUS (der bekannte Lotus in einer neuen Sequenz), WILKERSON, DURETT und RELIANT (Raumschiffe in schneller Rotation).

KICK 105: ANIMATIONEN

LLEWELLYN ist eine mit dem Sculpt 3D-Animator erstellte Filmsequenz mit musikalischer Untermalung. Eine schöne Demo dieses Programms von Michael Clinard.

KICK 104: ANWENDUNGEN

DIRECTORY MASTER V1.1: Dieses Programm bringt Ordnung in Ihre Diskettensammlung. Die Dateien jeder eingelegten Diskette werden automatisch in eine Liste übernommen. Diese kann dann auf komfortable und vielfältige Weise bearbeitet und verändert werden. Sehr viele Funktionen, einfache Bedienung, sehr leistungsfähig und schnell. Als Zugabe gibt es die jeweils aktuelle Liste des KICKSTART PUBLIC DOMAIN SERVICE.
MEMOPAD: ist ein echt nützliches Hilfsprogramm für vergeßliche AMIGA-Anwender. Hier können wichtige Termine und Daten festgehalten werden. Bindet man den Befehl CHECKMEMO in die Startup-Sequence ein, dann erinnert das Programm an diese Termine. Einfache Mausbedienung, sehr komfortabel und hilfreich.

KICK 103: GRAFIK

C-LIGHT: ein sehr einfach zu bedienendes, mausunterstütztes Raytracer-Programm. Die Einstellungen erfolgen über viele Regler und Schalter. **NEWJUGGLER**: ein Grafikprogramm, das den Eric Graham's JUGGLER innerhalb eines Bildes ablaufen läßt. Source in C; Autor: Peter Weiland, Wietze. **HAM'S**: einfaches Malprogramm für den HAM-Modus in AmigaBasic. Das Programm kann leicht geändert werden, da der Source-Code gut strukturiert ist. Autor: Henning Frommer, Vallendar

KICK 102: SPRACHEN

STONY BROOK PROLOG: auf dieser Diskette befinden sich die Source-Codes (Lattice C 4.0 und Prolog) des PD-Prolog-Systems von Diskette 101.

KICK 101: SPRACHEN

STONY BROOK PROLOG Version 2.3.2: Dieses PD-Prolog-System basiert auf einem System, das ursprünglich für UNIX-Rechner konzipiert wurde. Das System beinhaltet sowohl einen Interpreter als auch einen Compiler (bis zu 3000 LIPS). Der Sprachumfang entspricht weitgehend dem Edinburgh-Standard. Das System besitzt zusätzlich einige Besonderheiten z.B. dynamische Prädikateinbindung, Mischen von interpretiertem und compiliertem Kode, Makros und andere Hilfen. Inklusive ca. 100seitiger Dokumentation. Benötigt mindestens 1MB Speicher.

KICK 100: JUBEL-DISK

Auf dieser Diskette sind Programme zusammengefaßt, von den wir denken, daß sie für einen AMIGA-Besitzer unverzichtbar sind. Die Palette reicht von Kopierprogrammen, CLI-Hilfen bis zu Spielen. Lassen Sie sich überraschen - es wird sich lohnen!

KICK 99: SPIELE

BrainWork: Puzzle. *Autor: Timmy.* **Amiga-Wurm**: Tron-Variante; AmigaBasic. *Autor: Christoph.* **May Valley of the Aztec's**: ähnlich "Kaiser"; ABasic. *Autor: Matthias Hakuba* **Bauernskat**: Source in AmigaBasic. *Autor: Oliver Peter* **Breakout**: 99 Level mit Editor; Source in C. *Autor: Peter Händel.* **Kniffel**: bekanntes Würfelsspiel für 4 Personen oder gegen Computer; sehr schöne Grafik. Mausbedienung. *Autor: Michael Teistler.* **PACCIE**: Pacman-Variante in Basic.

KICK 98: SUPER-SOUNDS

KICKSTART SOUND DEMO II: sicherlich eine der besten Sounddemos, die es derzeit auf Disk gibt. 18 programmierte Musikstücke, aktiongeladener Sound, Stereo, sehr lange Spielzeit, kurze Ladezeit durch gepacktes Format, Sound-Slideshow. *Komponist: Tom Beuke*

KICK 97: UTILITY

TASKMON V0.4: zeigt die Zustände aller Tasks; Source in C. *Autor: P. Erpenbeck.* **FilterSwitch 2.0**: damit läßt sich der Tiefpaßfilter per Alt-Taste an- und ausschalten. **DMASwitch 2.0**: Bildschirmabschalter; Source in C. *Autor: Andreas Jung.* **MouseTool** (Fenster hervorholen, vergrößern, verkleinern, PALPatch (verändert Intui.-Lib., vergrößert alle Screens und Windows auf PAL-Größe), ChangeTaskPri, CLI4WB (CLI-Programme von der WB starten); alle Prg. mit Source (Profimat-Assembler). *Autor: Richard Englert, Fürth.* **RossiDress V1.0**: benutzerfreundliche Adressenverwaltung; AMIGABasic. *Autor: Jürgen Rogg.* **Milben**: kleiner Screenhack; Source in C. *Autor: Guido Appenzeller*

KICK 96: UTILITY

TurboBackup: komfortables und schnelles Kopierprogramm; kopiert auf bis zu 3 Laufwerke mit Verify.
MRBackup2.1: Harddisk-Backupprogramm, sichert auch überlange Dateien.
DMouse: Bildschirmschoner, Mausabschalter, Mausbeschleuniger, Fensteraktivator, Funktionstastenbelegung.
ARC: Programm zum Packen und Entpacken von Dateien. Hiermit können Programme platzsparend abgelegt werden (siehe PD 94/95).

KICK 95: ANWENDUNGEN

AnalytiCALC: professionelles Tabellenkalkulationsprogramm (Spreadsheet), das auch schon für MS-DOS, VAX und PDP-11 existiert. Das Programm arbeitet im Interlace-Modus und benötigt deshalb 1MB Speicher. Sehr leistungsfähig (18.000 * 18.000 Zellen usw.).
Wichtig: Die Dateien der Disketten 94 & 95 sind in komprimierter Form abgelegt. Sie benötigen das Programm ARC (PD 96), um sie wieder zu entpacken.

KICK 94: ANWENDUNGEN

RIM-5 ist ein relationales Datenbanksystem, für Verwaltung großer Datenmengen. Integrierte Programmiersprache. Help-Funktion. Incl. 80-seitiger Anleitung und Source in Absoft-Fortran .

PD 93: MODULA-2 (AMOK #6)

SoftScroll (zum Scrollen einer BitMap), MathLib (Bibliothek für Vektoren und Matrizen), IFFSupport (Laden und Speichern von ILBM-Bildern), ListAll (erweitert LIST für Subdirectories.

PD 92: MODULA-2 (AMOK #5)

Graphics (IFF-Bilder mit Mausklick), PrinterSupport, Print (TYPE TO PRT:), MemSystem (intelligenter Heap-Ersatz), Superlist1.2 (für Scrollbar-Requester), PrintIt! (Hardcopyprogramm), IconSupport (zum Erzeugen von Icons)

PD 91: MODULA-2 (AMOK #4)

CHECKER ist im Rahmen einer Studienarbeit im Fach Technisches Design entstanden. Thema war die "Erarbeitung neuer Möglichkeiten der Informationsdarbietung in Kraftfahrzeugen unter Zuhilfenahme eines microcomputergestützten Grafikbildschirms".

PD 90: UTILITIES

PERFECTSOUND V2.1: Ein umfangreicher Soundsampler mit vielen Features (z.B. Stereo samplen, Stereo erzeugen, Instrument erzeugen, Bereiche manipulieren, umkehren, löschen uvm.). Einfache Mausbedienung (siehe auch Bericht in KICKSTART 10.88).
CROSSREF: Dieses Programm erzeugt eine Aufstellung aller Variablen und Befehle mit Zeilenverweis. Unverzichtbar für Programmierer; verwendbar für alle ASCII-Dateien. M2-Modula. *Autor: Jörg Gutzke, Mönchengladbach*
DISKLAB: ein Programm zum Drucken von Disketten-Labels auf 9 Nadel-Druckern (Anpassung an andere möglich!). Mausgesteuerte Auswahl von Direktoryeinträgen, Grafikeinbindung, sehr komfortabel. Source in AMIGABasic. *Autor: Manfred Illi, Ludwigshafen.*
BOOTBLOCK-CHECKER: automatisches Testen des Bootblocks, akustischer Alarm, ASCII-Ausgabe u.a. *Autor: Matthias Kühn, Weinheim*

PD 89: DIGI-SOUNDS

The Sound of Music (Volume 3): weitere digitalisierte Musikstücke von *Harald Schneider*, mit Player. Hitparadensounds für eigene Programme und Vorspänne.

PD 88: MUSIK

Weitere Musikstücke von *Andreas Starr*, die man nicht verpassen sollte. Stereo, mit Player.

PD 87: MUSIK

Sehr schöne Musikkompositionen schickte uns *Andreas Starr* aus Gronau. Er hat sie mit Sonix erstellt. 5 Stücke (u.a. Miami Vice-Variation) in echtem Stereo mit einer Spielzeit von fast 20 Minuten. Die Musik wird im Hintergrund abgespielt. Lassen Sie sich dieses Erlebnis nicht entgehen.

PD 86: SPIELE

CYCLES 1.0: Eine sehr spannende Tron-Variante, bei der man sein Fahrzeug um Gegner und Quadrate herumlenken muß. In höheren Leveln wird die Geschwindigkeit größer, und immer mehr Quadrate erschweren die Suche nach einem freien Weg.

ESCAPE FROM JOVI: Wer wagt mit dem Raumschiff die Flucht durch die engen Gänge von Jovi? Source in Assembler. *Autor: Oliver Wagner, Sprockhövel*
ZAUBERFLÄCHEN: Erfordert gute Taktik, die verschiedenen Farbfelder zu sortieren. Kein Spiel für Leute, die gleich verzweifeln. Source in AMIGABasic. *Autor: Heiko Jahn*

PD 85: SPIELE

Eines der besten und spannendsten Strategiespiele ist **RISK V3.0**. Bis zu 5 Spieler müssen versuchen, die Herrschaft über die Welt zu erlangen. Dabei entscheidet nicht nur das Glück, sondern es kommt auch auf geschicktes Verhandeln an. Besonders wenn viele mitspielen, ist es schnell um einen geschehen, bzw. man wird nie sein Ziel erreichen. Das Spiel hat eine sehr schöne Grafik, ist sehr einfach mit der Maus zu bedienen und hat eine Vielzahl von Optionen, die den Spielablauf erleichtern oder erschweren können. Ein ungeheurer Spaß für lustige Spielerrunden. *Autoren: Michael Berling & Hartmut Stein (Bernstein Zirkel Softworks)*

PD 84: BILDER

Eine Bildershow mit fantastisch digitalisierten Bildern, Sound und Effekten. Die sehr realistischen und kunstvollen Aufnahmen zeigen was man, mit der entsprechenden Ausstattung, auf dem AMIGA machen kann. Erstellt mit dem DVS-2000 System von PBC Computerdesign.

PD 83: MODULA-2 (AMOK #3)

IFF-Loader in Assembler, Konverter für IFF-Brushes, Rekorder für Maus- und Tastenaktionen, Tool für GELs und anderes mehr.

PD 82: MODULA-2 (AMOK #2)

Die zweite Diskette des Modula-Klubs AMOK mit Routinen zu folgenden Themen: IFF-Loader, Halfbrite- und HAM-Demo, Hilfe zur Intuitionprogrammierung, Requesterroutinen.

PD 81: MODULA-2 (AMOK #1)

7 begeisterte Modula-Fans aus dem Raum Stuttgart haben sich zum Amiga Modula Klub (AMOK) zusammengeschlossen. Ihre Erfahrungen machen sie jedem Interessierten über die AMOK-Disketten zugänglich, auf denen sich hilfreiche Routinen in Form von Modulen befinden. Zu jedem Modul gibt es eine ausführliche Anleitung und ein Beispielprogramm. Auf der ersten Diskette befinden sich Programme wie z.B. Sprite- und Bob-Handhabung, Joystick-Routine, Interruptsteuerung und vieles mehr.

PD 80: UTILITIES

BOOTBLOCK CHAMPION testet den Bootblock auf Viren und erlaubt es, Assemblerprogramme darauf zu schreiben. Außerdem kann der Bootblock gesichert werden. *Autor: Roger Fischlin*

VIRUSTEST und **VIRUSCHECK** sind Virustestprogramme, die in die Startup-Sequence eingebunden werden können. *Autor: Roger Fischlin*

DFC ist ein einfach zu bedienendes, aber leistungsfähiges Kopierprogramm. Formatieren, Verify und Kopieren auf bis zu 4 Disketten sind die Optionen des Programms.

PD 79: SCHULE

PROCALC 1.1 ist eine komfortable Taschenrechnersimulation eines HP11G. Die Bedienung solcher Rechner ist einfach, jedoch anfangs gewöhnungsbedürftig (UPN). Der große Leistungsumfang wird in einer ausführlichen deutschen Anleitung erklärt. *Autor: Götz Müller*

GRAPH-MASTER V3: ein sehr umfangreiches Programm zum Darstellen von Funktionen. *Autor: Rolf Beck*

3-D FUNKTIONEN ist ein komfortabler und schneller 3D-Plotter. Source in AMIGABasic.

PD 78: GRAFIK-SHOW

FRAKTAL-MOVIE ist ein Film aus 200 s/w-Fraktalgrafiken, die hintereinander gezeigt werden (ca. 8 Bilder/s). Dabei wird aus einem Fraktal langsam herausgezoomt, bis man das nächste erkennen kann usw.. Benötigt für den kompletten Film 1MB Speicher! *Autor: Jürgen Weineit*

PD 77: SPIELE

KART ist ein Autorennspiel, bei dem zwei Spieler ihren jeweiligen Wagen über 6 verschiedene Rennstrecken lenken müssen. Höchst-, Lenkgeschwindigkeit und die Rundenanzahl können vorgewählt werden. Gute Grafik und Sounds; erfordert zwei Joysticks. Source in C. *Autor: Axel Ilenburg*
BUNDESLIGA ist ein Spiel, bei dem eine Fußballmannschaft zur Meisterschaft geführt werden soll. Geschicktes Kaufen und Verkaufen und die Aufstellung des Teams sind die wichtigsten Maßnahmen für den Erfolg. Source in AMIGABasic. *Autor: Ingo Hirsch*

PD 76: SPIELE

KAMPF UM ERIADOR V2.0 ist ein Fantasy-Strategiespiel für zwei Personen. Im Jahre 6735 wogt der Kampf zwischen Gut und Böse im Land Eglador. Auf einem großen Feld stehen sich die Legionen der beiden Seiten gegenüber. Mit strategischem Geschick müssen Sie versuchen, den gegnerischen Magier zu erreichen, der sich am anderen Ende des Spielfeldes in seiner Festung befindet. Gute Grafik und Sounds; benötigt 1MB Speicher!
Autor: Ralf Böwing, Roland Hartz

JETZT NUR NOCH
DM 8.-
PRO DISKETTE

Wir verwenden nur doppelseitige Markendisketten der Firmen JVC und MAXELL.

PD 75: SUPER-SOUNDS

KICKSTART SOUND DEMO: Nicht digitalisierte Sounds, sondern programmierte Musikstücke schickt uns Tom Beuke, damit wir "Abstand nehmen von den schnöden Digi-Sounds (kann doch jeder, oder?)". 17 actiongeladene Musikstücke; Länge 12-95kB. *Komponisten: TOB & TWP*

PD 74: DIGI-SOUNDS

The Sound of Musik (Volume 1): Digitalisiert Musikstücke zur Verwendung in Vorspännen, Demos und eigenen Programmen. Sehr gute Qualtität; mit Player; 11 Samples; Länge 30-100kB. *Sampler: Harald Schneider*

PD 73: SPIELE

LIGHT CYCLE 3D: ist eine sehr gut gemachte TRON-Variante, bei der zwei Fahrer versuchen, dem Gegenspieler den Weg abzuschneiden. Die 3D-Grafik ist dabei ein wirkliches Erlebnis. Da jeder Fahrer eine undurchdringbare Bahn hinterläßt wird das Spielfeld schnell eng. Mit vielen Anzeigeinstrumenten und Anleitung. *Autor: Tobias Richter*
ARTILLERIE-DUELL: in einer zerklüfteten Berglandschaft sehen sich zwei Kanonen gegenüber. Jeder der zwei Spieler muß nun den Abschußwinkel und die Stärke einstellen, mit der er seine Kugel abfeuern will. *Autor: Peter Kliem*
TREFFER: eine Kniffel-Variante, die komplett mit der Maus gesteuert wird. Bis zu vier Spieler, mit Zwischenstand, Hi-Score und Statistik. *Autor: Michael Jänecke*

PD 72: ANIMATIONEN

Für alle **STAR TREK**-Fans hat Tobias Richter zwei Animationen mit VideoScape 3D erstellt. Die erste zeigt ein Klingonenschiff, die zweite die U.S.S. ENTERPRISE.

PD 71: UTILITIES

PRTDRVGEN 2: der Druckertreibergenerator zum Anpassen der exotischsten Typen an den AMIGA. Sehr einfache Benutzerführung mit ausführlicher (englischer) Erläuterung aller Optionen (siehe Bericht in KICKSTART 8/9 88).
VIRUS X: läuft im Hintergrund und prüft den Bootsektor jeder eingelegten Diskette auf Viren und mögliche Veränderungen. Mit Source.
Overscan: ein Patch für die Intuition-Library, mit der amerikanische Programme das PAL-Format ausnutzen können, ohne daß sie modifiziert werden müssen.
HP: ein RPN-Taschenrechner mit vielen Funktionen (BIN, OCT, DEC, HEX, SIN, COS, TAN, ATN, LOG, LN, LG2, ATN, y^x ...) und 32 Registern. Mit Source in C

PD 70: SPIELE

WHEEL: Bei diesem Spiel muß ein Ausdruck erraten werden. Diese Glücksrad-Simulation ist grafisch gut aufgemacht und komplett mausgesteuert. Da die zu erratenden Begriffe in englisch sind, ist dieses Spiel gut zum Erlernen dieser Sprache geeignet. **MASTERMIND**: Kombinationsspiel, bei dem eine Farbkombination erraten werden muß. Mausgesteuert! **RISTINOLLA**: Eine spielstarke GO-Muko-Variante. Mausgesteuert und sehr schnell! **CLUE**: das bekannte Brettspiel für pfiffige Detektive in einer ansprechenden Computerversion. Verschiedene Räume müssen untersucht, Personen befragt und die Ereignisse kombiniert werden um den Mörder herauszufinden. Die gute Grafik und die einfache Mausbedienung machen eine Anleitung weitgehend unnötig.

PD 69: GRAFIK-DEMOS

AMUC_DEMO: Ein Bild mit 200 x 2000 Bildpunkten scrollt vertikal über den Bildschirm. Ein sehenswertes Erlebnis.
HBHILL: Dies ist Kevin Sullivans Beitrag zum BADGE KILLER DEMO CONTEST. Die Demo nutzt den Extra-Half-Brite-Modus (64 Farben) des AMIGA.
HAMmmm: Eine mit Musik unterlegte Linendemo von Phil Burk. Inclusive Source-Code in JForth. **STARS**: Ein interessanter Flug durch einen Sternenhimmel.

PD 68: GRAFIK-DEMOS

WBLANDER: versuchen Sie das Raumschiff sicher auf einem Workbench-Fenster zu landen. Auch dieses Programm hat am BADGE KILLER DEMO CONTEST teilgenommen

VERSANDBEDINGUNGEN:

Um einen schnellen und problemlosen Versand zu gewährleisten, beachten Sie bitte folgende Punkte:

- Bestellungen per Nachnahme oder Vorrauskasse
- Für jede Diskette ergibt sich ein Unkostenbeitrag von DM 8.-
- Pro Sendung kommt ein Versandkostenbetrag (für Porto und Verpackung) von DM 5.- (Ausland DM 10.-) hinzu.

Bei einer Bestellung von 5 oder mehr Disketten entfällt der Versandkostenbeitrag!

- Bei Nachnahme zuzüglich 3.70 DM

Anschrift:
MAXON Computer GmbH
KICKSTART PD
Postfach 55 69
6236 Eschborn

Die Diskettenbestellung kann auch telefonisch erfolgen. Der Versand erfolgt dann per Nachnahme.
Tel.: 06196 / 48 18 11
(Mo.-Fr. von 9^{00}-13^{00} u. 14^{00}-17^{00} Uhr)

KICKSTART PD 1-67

PD 67: UTILITIES

PR: ein Programm zum Ausdrucken von Texten. Das Inhaltverzeichnis wird in einem Fenster dargestellt und das File kann mit der Maus ausgewählt werden. Auf dem Ausdruck erscheint dann eine fortlaufende Seitennummerierung, der Name und das Datum. **FUEL GAUGE:** ist eine grafische Anzeige für den vorhandenen und benutzen Speicherbereich. **SIMCPM**: CP/M-Emulator, der den 8080-Prozessor emuliert. Mit H19 Terminal. **LOGO:** ein Interpreter der bekannten Turtle-Spache, die mehr kann als nur malen. **MICROSPELL:** ist ein flexibler Spellchecker mit einem englischen Grundwortschatz von 43.000 Wörtern.

PD 66: M2-Modula

M2 AMIGA: Dies ist die funktionsfähige Demo-Version des M2-Modula-Compilers. Beschränkungen gibt es nur hinsichtlich der Größe der compilierbaren Programme. Auf der Diskette sind einige Demo-Programme enthalten, die z.T. direkt für diesen Compiler entwickelt wurden.

PD 65: ICON-TOOLS

IDPICON: ermöglicht es, eine mit Deluxe Paint erstellte Brush beliebiger Größe in ein Icon umzuwandeln.
SNAP: wandelt ein beliebiger Bildschirm in ein Icon um. Die Größe des Icons ist dabei einstellbar. Bei mehrfarbigen Screens wird eine Farbreduzierung durchgeführt.
ALTICON: gibt dem eigenen Icon den letzten Pfiff, denn hiermit werden zwei Icon-Bilder zu einem 'animierten' Icon zusammengefügt. Außerdem viele schöne Icons.

PD 64: (Strategie-) SPIELE

BACKGAMMON: bei diesem Strategiespiel können Sie zeigen, ob Sie besser sind als der Computer. **PUSH OVER:** hier wird so lange geschoben bis plötzlich und unerwartet einer der Spieler fünf Steine in einer Reihe hat. Ein spannendes Strategiespiel. **PUZZLE PRO, IFF2PICS:** zwei Programme, die ein beliebiges IFF-Bild in viele Einzelteile zerlegen. Dann liegt es an Ihnen, wie lange Sie brauchen, um es wieder zusammenzupuzzeln. Einstellbarer Schwierigkeitsgrad.

PD 63: (Action-) SPIELE

AMOEBA: hinter diesem Namen verbirgt sich eine spielhallenmäßige SPACE INVADERS-Adaption. Obwohl dies ein Spiel der ersten Computergeneration ist, hat es nichts von seiner Spannung verloren. Ein sehr gutes Ballerspiel. **ASTEROIDS** ist ein Weltraumspiel, das ebenfalls von Spielhallencomputern umgesetzt wurde. Es ist nicht ganz einfach, die Kontrolle über das Raumschiff zu behalten und es sicher im Asteroidenhaufen zu manövrieren. **EGYPTIAN RUN**: Actionspiel, bei dem ein Geländewagen durch die Wüste gesteuert wird.

PD 62: GRAFIK-DEMOS

HAGENDEMOS: Joel Hagens Film "RGB" ist der Gewinner des BADGE KILLER DEMO CONTEST (benötigt 1 MB RAM !). **BULLY**: Diese Demo ist der absolute Wahnsinn. Drei Bildschirme mit Animationen sind auch noch in Bewegung. Da zeigt der AMIGA was Multitasking bedeutet.

PD 61: GRAFIK-DEMOS

RIPPLES: eine eindrucksvolle Sequenz, die mit VideoScape 3D erstellt wurde. **WAVEBENCH** bringt Ihren Bildschirm gewaltig ins Wanken; **VIACOM** ist auch nicht viel besser! **DROPSHADOW 2.0:** von nun an hat jedes Fenster eine Schattenseite, mit Einstellregler für Schattenstärke und -länge.

PD 60: RAY-TRACER

A-Render ist ein leitungsfähiges Programm zum Erstellen von Bildern nach dem Ray-Tracing-Verfahren. Dateneingabe per Texteditor. Die Berechnung der Bilder durch das Programm kann, wegen des aufwendigen Verfahrens, mehrere Stunden dauern, aber die Ergebnisse sind hervorragend.

Diskette 59: Grafik-Animation

Mit drei herrlichen Grafikdemos stellt Eric Graham seine Programme Sculpt 3-D und Animate 3-D vor. Die Bilder der Animationen sind nach dem Ray-Tracer-Verfahren berechnet.

Diskette 58: Grafik-Animation

Demos vom BADGE KILLER DEMO CONTEST. Hervorragende Grafik und Animation.

Diskette 57: UTILITIES

GOMF (bekämpft den GURU), VCHECK (Virustester von Commodore), JOURNAL (Mausaktionen aufnehmen und abspielen), PRINTERSTEALER (Druckerausgabe auf Diskette umleiten), Editordemo

Diskette 55: Grafik/Utilities

Einige schöne Grafikdemos und Utilities zu diesem Thema

Diskette 54: Anwenderprogramme

MICROSPELL (überprüft die Rechtschreibung), ACCESS (Terminal), QBASE (Dateiverwaltung) uvm.

Diskette 52: UTILITIES

CONMAN V1.0 (sehr nützlicher CLI-Ersatz), MOUSEREADER (Texte lesen mit Mausbedienung), TIMERAM, MEMWATCH, **DISKMAN V1.0** (Bedienung fast aller Diskettenoperationen mit der Maus, sehr umfangreich!), DIRMASTER V1a (sehr schönes Disketten-Verwaltungsprogramm, viele Funktionen)

Diskette 51: C-Compiler

Ein einfacher C-Compiler, in den man etwas Arbeit stecken muß, damit er läuft. Eignet sich für Interessenten am Compilerbau, da der Source-Code vorliegt.

Diskette 50: BASIC

Eine Diskette voll Programmen (Spiele, Grafikprogramme uva.) in AMIGABasic zum Reinschauen, Verändern, Lernen.

Diskette 49: ICONS

Utility-Programme, die sich mit der Erstellung und Manipulation von Icons beschäftigen. z.B.: XICON 2.0 (mit diesem Programm können Dateien ausgeführt werden, die CLI-Kommandos enthalten)

Diskette 48: CRAZY

Auf dieser Diskette befinden sich nur verrückte Programme, deren Sinn absolut zweifelhaft ist. Allerdings sollten Sie sich diesen Spaß nicht entgehen lassen!

Diskette 47: UTILITIES

SECTORAMA: ein sehr nützlicher Disketten- und Festplatten-Monitor, mit dem verlorene oder zerstörte Daten wiederhergestellt werden können. **SILICON:** sehr komfortabler CLI-Ersatz mit separatem Ausgabe- und Eingabefenster. **DBUG**: maschinenunabhängiger Debugger von Fred Fish (Source in C) **TIMER**: Stoppuhr für die Workbench

Diskette 46: GRAFIK-SHOW

Eine weitere Diskette (siehe auch PD 42) mit phantastischen RAY-TRACER-Bildern, unterlegt mit digitalisierter, fetziger Musik.

Diskette 45: SPIELE

GRAVITYWARS ist ein interessantes Weltraumspiel, bei dem sich zwei Raumschiffe im Kampf gegenüberstehen. **OTHELLO**, eine sehr schöne Reversi-Variante (mit eigenem Fenster!). **STREITPATIENCE**, eine Patience-Variante von Hellmut Voelcker (Berlin) **CHESS**: spielstarkes Schachprogramm. **ADVENTURE**: Textadventure

Diskette 44: SPIELE

Adventurefans kennen sicherlich das Grafikadventure HACK (siehe PD 3). Hier die Fortsetzung: LARN. In unüberschaubaren unterirdischen Gängen müssen Gold und Schätze gesucht werden. Aber auch Krafttrunke oder magische Sprüche sollte man nicht ignorieren, denn die benötigt man im Kampf gegen Gnome, Vampire und andere Gestalten. Wirklich sehr empfehlenswert!

Diskette 43: GRAFIK-SHOW

Eine einmalige Show, bei der eine digitalisierte Katze in gleitenden Bewegungen über den Bildschirm trabt. Estellt wurde diese faszinierende Animation mit einem Digitizer, DPaint und VideoScape 3D.

Diskette 42: GRAFIK-SHOW

Vielfältige nach dem RAY-TRACER-Verfahren erstellte Bilder. Lassen Sie sich von den realistischen Spiegelungen beeindrukken! Mit digitalisierter Musik!

Diskette 41: UTILITIES (Grafik)

Alles, was Sie zu dem von ELECTRONIC ARTS entwickelten Grafik-Standard (IFF-Format) wissen müssen: Laden, Speichern, Komprimieren, Dekomprimieren. Mit Dokumentationen und Source-Codes in C.

Diskette 40: GRAFIK-DEMOS

Boing!, Rotate, Sparks, Moire, Dazzle, 3DCube, Scales, Sizzlers. Sehenswert ist der Film 'Atari meets AMIGA', wer wohl gewinnt? Sehr schön ist das Programm LANDSCAPE, das wunderschöne fraktale Berg- und Tallandschaften erzeugt.

Diskette 39: GRAFIK-SHOW

Stimmungsvolle Landschaftsbilder, die sich gut zum Weiterverarbeiten eignen, und einige digitalisierte Bilder.

Diskette 38: GRAFIK

NoFFP Mandelbrot Set Explorer V2.1 (neue Version) von ABC Softarts in Braunschweig.

Diskette 37: UTILITIES

AddMem (zum Konfigurieren von Speichererweiterungen), MemView (zeigt den Speicherinhalt als Grafik an), GetRom (schreibt das Betriebssystem-ROM des AMIGA 500/2000 als bootfähige Kickstart für den AMIGA 1000 auf Diskette), MegaPatch (paßt die Kickstart des AMIGA 1000 für das autom. Erkennen von internen Speichererweiterungen an)

Diskette 36: CAD

mCAD ist ein wirklich gut gemachtes CAD-Programm, daß jedoch nur im Interlace-Modus läuft. Es bietet die einfachen Zeichenfunktionen und Features wie Zoom, Group, Ungroup, Grid, Move, Rotate). Auf der Diskette sind mehrere Dokumente, die das Programm erklären.

Diskette 35: UTILITIES

ASDG (resetfeste RAM-Disk), FixDisk, ErrorCk (zur Fehlersuche auf der Diskette). **DiskCat** (erstellt eine Übersicht über die Programme Ihrer Disketten)

Diskette 34: SPIELE

TUNNEL VISION: Werden Sie den Weg durch das Labyrinth finden? **REVERSI**: eine spielstarke Version des bekannten Brettspiels. **KLONDIKE**: Patience-Kartenspiel

Diskette 30: SOUND-DEMOS

Digitalisierte Songs: Changing Minds, Joan Lui, Miami Vice II, Respactable, Holiday

Diskette 26 & 27: Grafik-Show

Auf zwei randvollen Disketten erleben Sie eine einmalige Dia-Show mit hervorragend digitalisierten futuristischen Bildern in voller PAL-Auflösung. Dazu gibt es stimmungsvolle, sphärische Musik.

Diskette 25: UTILITIES

CLOCK, PORTAR, MACView, Kickbench, Disassembler, Tracker, Checkmodem, POPCLI und vieles mehr

Diskette 24: Grafik-Show

Sehr schöne, digitalisierte Frauengesichter.

Diskette 23: Grafik-Show

Viele abwechslungsreiche Motive in verschiedenen Auflösungen, mit Grafik-Show.

Diskette 22: Sprachen

C-FORTH (recht leitungsfähiger FORTH-Interpreter, der auch als Quelltext vorliegt)

Diskette 21: AMIGA-Tutor

Einführung in die Bedienung des AMIGA 500. Ein farbenfroher Lehrgang, der ganz am Anfang beginnt und mit vielen Bildern und Grafiken die Grundbegriffe des AMIGA erklärt. (für Anfänger, komplett in deutsch)

Diskette 19: Grafik-Show

Sehr schöne digitalisierte H.A.M.-Bilder

Diskette 16: Sprachen

XLISP 1.7 (neueste Version) mit ausführlicher Anleitung (über 50k)

Diskette 14: EDITOR

Bekannter Texteditor MICROEMACS Version 30 (viele Features: Search/Replace/Copy)

Diskette 11: Grafik-Show

RAY TRACERS: wunderschöne räumliche Bilder, die auf einer VAX berechnet und auf den AMIGA übertragen wurden

Diskette 10: Grafik-Show

JUGGLER DEMO: ein bewegliches Männchen jongliert mit drei verspiegelten Kugeln, sehr schöne Demo

Diskette 9: Grafik-Show

Grafik-Show mit bekannten Cartoons und schönen Landschaftsbildern

Diskette 8: Spiele

Monopoly: das bekannte Brettspiel mit sehr schöne Grafik, einfache Mausbedienung, bis zu vier Spieler (Source in ABasic)

Diskette 7: UTILITIES

QuickCopy (gutes Kopierprogramm), **DirUtil** (File-Copy), **FileZap** (File-Monitor), **DiskZap** (Disk-Monitor), **DiskSalv** (Diskettenretter), **System-Monitor, CSH** (UNIX-ähnliche Shell)

Diskette 5: Terminal-Programme

WOMBAT (VT102/52 Emulator, XModem, autodial), VT100 (grafikfähig, Soure in C), TermPlus (XModem, Soure in C), DG210 (Data General D-210 Emulator), Ahost (XModem, Kermit), TEK4010 (XModem, VT100)

Diskette 3: Spiele

HACK: das bekannte Textadventure, das ursprünglich auf UNIX-Rechnern erstellt wurde, liegt hier als spezielle Grafikversion für den Amiga vor.

Diskette 2: Spiele

YachtC (Würfelspiel für 4 Personen), Puzzle, Missile (verteidigen Sie Ihre Stadt, starker Sound), TriClops (sehr schönes 3D-Spiel!), Breakout (3D-Effekt mit Brille), Trek73 (bekannte Star Trek-Variante)

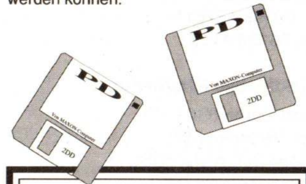
Diskette 1: C-Source

Eine Sammlung von Programmen, die besonders dem Anfänger zeigen, wie man Intuition programmiert. Die Programme liegen sowohl aus C-Quellcode als auch als fertige Programme vor, die sofort gestartet werden können.

KICKSTART PD-SAMMLUNG:

Die einzelnen Disketten werden nach festen Kriterien zusammengestellt, d.h. daß jede Diskette einen Schwerpunkt hat (z.B. Lehrgänge (Tutor), Bilder-Show, C-Programme, Utilities, Spiele, u. ä.). Außerdem werden Angaben über die Programmiersprache, den verwendeten Interpreter oder Compiler usw. gemacht.

Die Programme laufen auf allen AMIGA-Computern mit Kickstart/Workbench 1.2, allerdings sollten mindestens 512k Speicher vorhanden sein, die meisten Grafikdemos benötigen sogar 1MB Speicher. Sollten dennoch Einschränkungen gelten, wird dies bei den betreffenden Programmen angegeben.

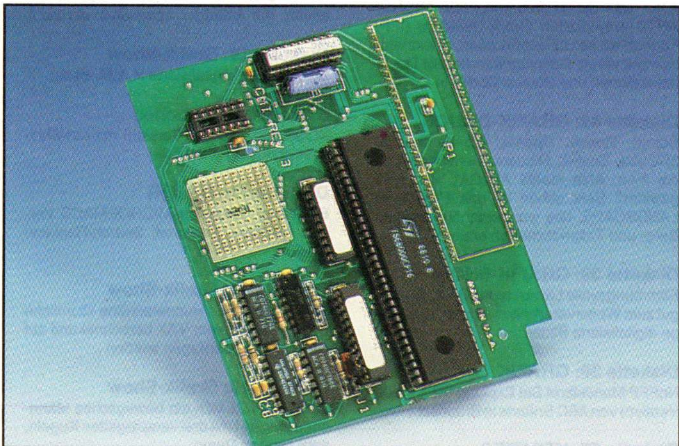


ANRUF GENÜGT
06196 / 481811

VORSCHAU

68000-Prozessor mit 14MHz kontra 68020 mit 68882-FPU-Platine

Im Vergleich stehen eine kleine 68020-Platine und die weit billigere Lösung des normalen 68000-Prozessors, der mit 14MHz getaktet wird. Reichen die Leistungsdaten des 68000 aus, um mit dem 68020 gleichzuziehen, oder gerät er doch ins Hintertreffen? Einen Vergleichstest lesen Sie in der April-Ausgabe.



SCULPT 4D versus TURBO SILVER 3.0

Zwei Raytracing-Giganten im Test: Das nagelneue Programm Sculpt 4D soll neue Maßstäbe setzen, Turbo Silver 3.0 glänzt bereits mit schnellen Berechnungszeiten. Doch welches Programm ist besser zu bedienen und leistungsfähiger? Die nächste KICKSTART verrät, welches Programm schließlich und endlich die NASE VORN hatte.

Der AMIGA-TOWER

Was IBM kann, können wir schon lange. So oder ähnlich dachten wohl ein paar Entwickler und entwarfen den TOWER-AMIGA. Natürlich wartet der TOWER auch mit besseren Leitungsdaten auf als ein normaler AMIGA. Grundbaustein ist eine A2000B-Platine. Was der Anwender vom TOWER-AMIGA erwarten kann, lesen Sie in der nächsten Ausgabe.

Ab 17. März an Ihrem Kiosk.

Änderungen vorbehalten

Impressum

KICKSTART

Chefredakteur:
Uwe Bärtels (Chefredakteur)(UB)
Markus Nerding (Stellvertreter) (MN)

Redaktion:
Andreas Krämer (AK)
Harald Schneider (HS)
Marcelo Merino (MM)
Harald Egel (HE)
Christian Keller (CHK)

Herausgeber:
MAXON-Computer GmbH
Industriestraße 26
Postfach 5569
6236 Eschborn
Tel.: 06196/481811
FAX: 06196/41137

Redaktionelle Mitarbeiter:
Gerald Carda (GC)
Jobst Hermeier (JH)
Carsten Borgmeier (CBO)
Martin Silbernagl (MS)
Sven Stillich (SS)
Florian Du Bois (FDB)
Sebastian Dosch (SD)
Wolf Dietrich (WD)

Public Relations:
Claus Peter Lippert

Auslandskorrespondenz:
D. Dela Fuente (ddf/GB)
L. Hennely (USA)

Verlag:
Heim Verlag
Heidelberger Landstraße 194
6100 Darmstadt 13
Tel.:06151/56057
FAX:06151/55689
06151/56059

Verlagsleitung:
Hans-Jörg Heim

Anzeigenverkauf:
Kyriakulla Margaritis
Uwe Heim (Ltg.)

Anzeigenpreise:
nach Preisliste Nr.3, gültig ab 1.1.88

Fotografie, Titelbild:
Klaus Ohlenschläger Foto & Design

Illustrationen:
Barbara Jacobs

Produktion:
Karl-Heinz Hoffmann

Druck:
Ferling Druck W.Frotscher GmbH, Darmstadt

Bezugsmöglichkeit:
Zeitschriftenhandel, Kauf- und Warenhäuser, Commodore-Fachhändler oder direkt beim Verlag.
KICKSTART erscheint 11 mal im Jahr

Einzelpreis: DM 7,- ,ÖS 56,- SFr 7,-
JahresabonnementInland: DM 70,-
Europ. Ausland DM 90,-
Luftpost DM 120,-

In den Preisen enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren.

Wir bedanken uns bei:

Josch, dem feucht-fröhlichen Balanceur

Profilaufwerk 3,5"
Metallgehäuse ● einstellbare Lauwerknummer mit Displayanzeige ● digitale Trackanzeige ● Write Protect am Laufwerk schaltbar ● abschaltbar ● durchgeschleifter Bus
1 Jahr Garantie
Super ALCOMPreis **329,-**

Laufwerk 5,25"
40/80 Track ● Laufwerksbus durchgeschleift ● abschaltbar ● einstellbare Adressen ● MS- DOS- kompatibel ● mit Diskchange

Super ALCOMPreis	298,-
HD 1,6 MB (umschaltbar)	318,-
Amigafarbene Blende	+10,-
Write Protect Schalter	+15,-

Gemischtes Doppel 3,5/5,25"
einzeln ein-/abschaltbar ● einstellbare Laufwerksnummern mit Anzeige ● durchgeschleifter Bus ● bei 5,25" 40/80 Tracks umschaltbar ● Metallgehäuse ● 1 Jahr Garantie
Super ALCOMPreis **548,-**

500er Speichererweiterung
Für 512k zusätzliches RAM ● alle RAM-s gesockelt ● selbstkonfigurierend ● abschaltbar ● Uhrenschaltung auf Platine mit Akku- bzw. Batteriepufferung nachrüstbar

Komplett mit 512k	Preis auf Anfrage
Superpreis mit Uhr	Preis auf Anfrage
Bauteilesatz für Uhr ohne Akku	24,-
Leerplatine mit Stecker	*39,-

*mit Schaltplan und Bestückungsliste

Laufwerkanschlußkabel
Zum Anschluß von Laufwerken an alle Amigas ● mit Ansteuerelektronik

Für 3,5" Laufwerk	39,-
Für 5,25" Laufwerk	49,-

Steckplatzerweiterung
3-fach für Laufwerke
Jeder Steckplatz abschaltbar und einstellbare Laufwerksnummer ● Steckplatzerweiterung direkt am Amigagehäuse ● Dadurch keine Kabellängenprobleme
Anschlußfertig zum Super ALCOMPreis **49,-**

Soundsampler
Für alle Amiga's mit Software ● Type bei Bestellung bitte angeben ● 8-Bit Datenbreite ● Betrieb am Parallelport (Druckerport) ● Mit Vorverstärker für Micro-Anschluß (Cinch-Buchsen) ● Musik- und Sprachdigitalisierung möglich ● Arbeitet mit fast allen Digitizer-Programmen ● Formschönes Gehäuse
Super ALCOMPreis **79,-**

Sampler Studio
● Professionelles Sampler-Programm ● 4-Kanal-Technik ● speichern auf 4 Disketten hintereinander möglich ● alle gängigen Formate (IFF, Data, Future) ● Echtzeitdisplay mit Zoomfunktion ● viele Verfremdungsmöglichkeiten ● Echo, Hall, Reverse **69,-**
Paket: Sampler + Software **129,-**

MIDI - Interface
4 Kanäle einschließlich 1 Thru ● Optische Datenanzeige ● Formschönes Gehäuse
Wahnsinnspreis von nur **89,-**

Kickstartumschaltung
Bauen Sie die anderen Kickstart-Versionen in Ihren Amiga 500 ● Einfacher Einbau ohne Löten ● für Original-Kickstart-ROM und 2 zusätzliche Versionen auf EPROM ● EPROM-Programmierservice auf Anfrage

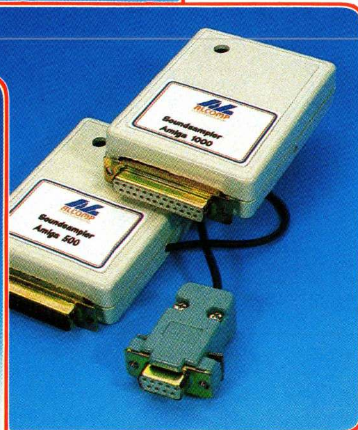
SuperALCOMPreis	59.-
Kickstartversion auf EPROM's	120.-

Userport + Experimentierkarte für Expansionport
Mit Lochraster und 2 x 6522 Ports

Leer	59.-
komplett aufgebaut	89.-

Wir suchen ständig Hardware-Entwicklungen. Wir garantieren gute Umsatzprovisionen und ehrliche Abrechnung

ausgereifte Ingenieurleistung ● 14 Tage Umtauschrecht ● fast alle IC's gesockelt ● nur professionelle Leiterplatten ● Bauteile namhafter Hersteller ● mit Bedienungsanleitung



kostenloses Info anfordern!!!

Bestellung und Versand

ALCOMP GmbH
Glescher Weg 22
5012 Bedburg
Tel. 0 22 72/20 93

Nachnahmeversand NN-Spesen 7.50 DM b. Vorkasse 3.- DM. Auslandsbestellungen: Nachnahmeversand NN-Spesen 10.- DM b. Vorkasse 5.-DM. Wir liefern Ihnen auf Ihre Rechnung und Gefahr zu den Verkaufs- und Lieferbedingungen des Elektronikgewerbes.
Postgiroamt Köln
(BLZ 370 100 50) 275 54-509

3,5" Laufwerk
Für alle Amiga's ● einstellbare Gerätenummer ● abschaltbar ● Metallgehäuse ● superflach ● 1 Zoll (2,54cm) ● durchgeschleifter Bus ● TEAC Laufwerk
1 Jahr Garantie

komplett anschlußfertig	239,-
Amigafarbene Blende	+10,-

Basislaufwerke 1 Jahr Garantie

TEAC FD 135 FN 3,5" 1MB superslimline	218,-
1,6 MB Diskchange	
Amigafarbene Blende	+10,-
3,5" Gehäuse	25,-
5,25" Gehäuse	25,-
Gehäuse für "Gemischtes Doppel"	65,-

Bootselector **19,90**

Amiga Eprommer
● Für A 500/1000
● Expansionsportanschluß
● Für EPROM's 2764-27011 (8K-128K) Alle A-Typen und CMOS-Typen
● Funktionen:
LEERTEST LADEN VON DISK
VERGLEICHEN SPEICHERN AUS DISK
AUSLESEN HEXDUMP
BRENNEN
● vier Programmieralgorithmen 50mS/Byte - Superschnell 64K-1,5 min
● Programm zum Generieren und Brennen von Kickstarts direkt von Diskette oder aus ROM
● Mit Software + Gehäuse **225,-**

Meß- und Steuerinterface
● 8 ADC-Kanäle 0-2,55V in 0,01V Stufe
● 1 DAC-Kanäle 0-2,55V in 0,01V Stufe Genauigkeit- 1,5 LSB
● 8 frei programmierbare TTL-I/O Kanäle
● Mit Gehäuse, Anschlüße auf Schraubklemmen
● interne Referenzspannung
● Expansionsanschluß
● Einfache Programmierung in Basic möglich Multitasking tauglich
● incl. DEMO-Software auf 3,5" Diskette
239,-

Trackanzeige
Für DFO-DF3 einstellbar ● für alle Laufwerke (3,5"/5,25") ● Laufwerkbus durchgeschleift ● mit Gehäuse
Super ALCOMPreis **69,-**

Einführungsangebot
Amiga - Harddisks
komplett anschlußfertig

Platte 20 MB A 2000	798,-
30 MB A 2000	898,-
40 MB A 2000	1 098,-
65 MB A 2000	1 348,-
Platte A 500/A 1000	
20 MB	898,-
30 MB	998,-
40 MB	1 248,-
65 MB	1 498,-
für den Selbstbau	
HD-Interface A 2000	198,-
HD-Interface A 500/A 1000	249,-

Vokabeltrainer
2500 englisch-deutsche Vokabeln incl. Hilfssatz ● Merkfunktion ● komfortabler Editor zur Vokabelverwaltung ● Wörterbuch zum Dateiendurchsuchen
59,-

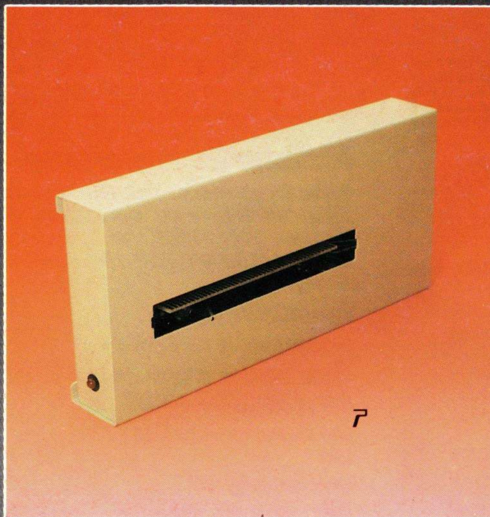
Selbstbootende Harddisk für Amiga ohne PC-Karte!
Die Amiga-Festplatte von ALCOMP:
● Selbstbootend wie "Card" oder "Rad"! ● Als Einbau-Festplatte für den "Amiga 2000" ● Als Externe Einheit für den "Amiga 500" und 1000 mit Gehäuse, eigenem Netzteil und Erweiterungsanschluß ● Erhältlich mit 20, 30, 40 und 65 Megabyte ● Kopiert 1 Megabyte in unter 4 Sekunden ● Speichert schneller als "1.2-Ramdisk" ● Läuft mit "FastFileSystem" ● Einfach einstecken, Formatieren, "Mountlist" und "Startup-Sequence" ändern und los geht's!
Entwickler: Stephan und Stefan
Für den Selbstbau: Harddisk-Interface incl. Steuersoftware ● Anschluß mit Slot für Omti-Controller

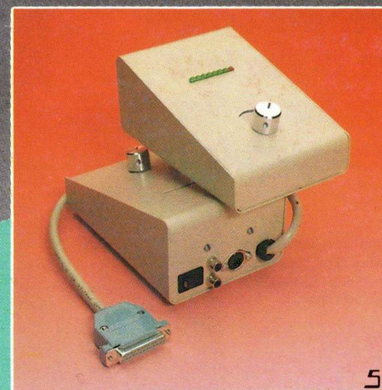
Entwurf + Druck PLUM, Düren-Oberzier, TEL. 0 24 28/7 11

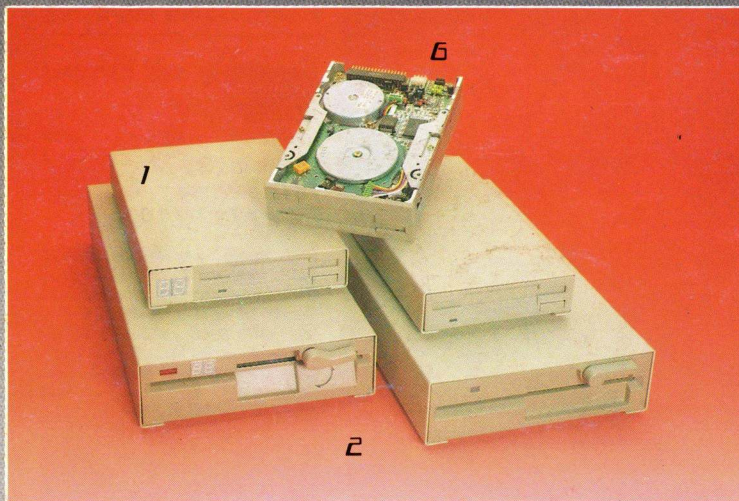
KUPKE

02 31/81 83 25-27
Telefax 02 31/81 74 29
D-4600 Dortmund 1
Burgweg 52 a

GOLEM







Wir liefern im 3-Tage-Rhythmus

1 Golem Drive 3,5 Display
NEC 1037a mit heller Frontblende ● Amiga-farbenes Metallgehäuse ● Abschalter ● Busdurchführung bis DF3 ● Sidecar, PC 1 und PC Karten-kompatibel ● Trackdisplay zur aktuellen Spur- und Kopfanzeige

mit **Display**	**DM 359.-**
ohne **Display**	**DM 339.-**

2 Golem Drive 5,25 Display
NEC Laufwerk mit heller Frontblende ● Amiga-farbenes Metallgehäuse ● Abschalter 40/80 Track-Umschalter ● Busdurchführung bis DF3 ● PC Karten, Sidecar und PC 1 kompatibel ● Trackdisplay zur aktuellen Spur- und Kopfanzeige

mit **Display**	**DM 449.-**
ohne **Display**	**DM 419.-**

3 Golem Ram Box 2MB
2MB Speichererweiterung für dem Amiga 1000 ● ansteckbar am Systembus ● Abschalter ● Busdurchführung ● autokonfigurierend ● Betriebskontrollanzeige ● Amiga-farbenes Metallgehäuse ● erweitert den Grundspeicher auf 2,5 Megabyte

komplett	**DM 1398.-**
ohne Ram's	**DM 449.-**

4 Golem 500 Ram Box
2MB Speichererweiterung im formschönen 500'er Design ● Busdurchführung ● autokonfigurierend ● Betriebskontrollanzeige ● externer Anschluß an den Systembus ● erweitert den Grundspeicher auf 2,5 Megabyte

komplett	**DM 1398.-**
ohne Ram's	**DM 449.-**

Technische Änderungen vorbehalten

5 Golem Sound Stereo
Audio Digitizer der Spitzenklasse ● kompatibel zur meisten Samplersoftware ● DIN- und Cinch Anschluß auch für Micro geeignet ● optisches Aussteuerungsdisplay ● Stereowandlung ● umschaltbar auf Mono-Betrieb

Stereo	**DM 189.-**
Mono ohne Display	**DM 139.-**

6 Golem Drive A 2000
internes Amiga Drive ● NEC 1036a mit heller Frontblende ● einbaufertig modifiziert ● mit Staubschutzklappe ● incl. Einbauanleitung und Montagesatz **DM 200.-**

7 Kickstart/Uhrenmodul
"Bitte Workbench einlegen", meldet ihr Amiga 1000 nach dem Einschalten mit dem extern ansteckbaren Kickstartmodul ● Busdurchführung ● Abschalter, so daß andere Kickstartversionen wieder gebootet weden können ● alle gängigen Kickstart-Versionen lieferbar **DM 199.-**

Amiga 500/2000 kompatibeles Uhrenmodul ● Akkugepuffert ● extern ansteckbar

im Extragehäuse	**DM 149.-**
Uhr u. Kick in einem Gehäuse	**DM 299.-**

8 Kickstartumschaltplatine
Intern einsteckbare Umschaltplatine bestückt mit einem zusätzlichen Kickstart ● alle gängigen Versionen ● keine Lötarbeiten erforderlich ● umschaltbar auf original Kickstart

komplett	**DM 149.-**
ohne Eprom's	**DM 59.-**

Entwurf + Druck PLUM Düren-Oberzier, TEL. 02428/7 11